

Affiliate B2C systém v technologii Ruby on Rails

Affiliate B2C system in Ruby on Rails technology

Souhlasím se zveřejněním této diplomové práce dle požadavků čl. 26, odst. 9 *Studijního a zkušebního řádu pro studium v magisterských programech VŠB-TU Ostrava*.

Práce byla vytvořena pro firmu Skvělý.CZ s.r.o. Veškerá data jsou smyšlená, zdrojová data jsou chráněna dle autorského zákona č. 121/2000 Sb.

V Ostravě 5. května 2010

.....

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 5. května 2010

.....

Rád bych na tomto místě poděkoval všem, kteří mi s prací pomohli, protože bez nich by tato práce nevznikla.

Abstrakt

Úkolem této práce je navrhnout a poté implementovat jádro affiliate internetového obchodu pomocí technologie Ruby on Rails. Nezbytnou součástí této práce bude seznámení se s pojmem affiliate marketing. V textu jsou vysvětleny základní principy affiliate marketingu, srovnána situace u nás a v zahraničí a poté popsány některé affiliate programy. Další část se bude věnovat frameworku Ruby on Rails, který je použitý při implementaci. Předposlední část čtenáři popisuje specifikace požadavků kladených na systém a poté analýzu těchto požadavků. Posledním bodem bude část návrhu a samotná implementace.

Klíčová slova: Affiliate marketing, Ruby, Ruby on Rails, MVC architektura

Abstract

The main objective of this work is to design and then implement a core affiliate online store by using Ruby on Rails technology. Essential part of this work will become familiar with the concept of affiliate marketing. The text explains basic principles of the affiliate marketing by comparing the situation here and abroad and then describe some affiliate programs. Another part will focus on Ruby on Rails which is used during the implementation. The penultimate section describes the specifications of the requirements on the system and then analyze these requirements. Last point is part of the design and actual implementation.

Keywords: Affiliate marketing, Ruby, Ruby on Rails, MVC architecture

Seznam použitých zkratek a symbolů

HTML	– Hyper Text Markup Language
XML	– eXtensible Markup Language
URL	– Uniform Resource Locator
B2C	– Business to Customer – označí pro obchodní vztahy mezi obchodními společnostmi a koncovými zákazníky, realizované webovými aplikacemi, virtuálními obchody na Internetu apod.

Obsah

1	Úvod	6
2	Affiliate marketing	8
2.1	Úvod do affiliate marketingu	8
2.2	Počátky affiliate marketingu	8
2.3	Typy affiliate programů	9
2.4	Provize pro affiliate partnery	9
2.5	Přínos affiliate pro obchodníky	10
2.6	Přínos affiliate pro provozovatele webů	11
2.7	Affiliate marketing a e-shopy	11
2.8	Affiliate marketing: situace v zahraničí	13
2.9	Affiliate marketing: situace na českém trhu	17
3	Použité technologie	22
3.1	Ruby	22
3.2	Ruby on Rails	23
3.3	MySQL	26
3.4	Javascript	27
3.5	AJAX	27
3.6	XHTML	27
3.7	CSS	27
4	Specifikace zadání	28
4.1	Vstupy	29
4.2	Výstupy	29
5	Specifikace požadavků	30
5.1	Specifikace pomocí případů užití	30
5.2	Specifikace chování	38
6	Analýza a návrh	42
6.1	Datová analýza	42
6.2	Statická struktura	42
6.3	Dynamická struktura	45
6.4	Model-View-Controller	46
6.5	ActiveRecord	46
6.6	Bezpečnost	47
7	Implementace	48
7.1	Struktura kódu pro import a export dat	48
7.2	Import zboží	48
7.3	Export zboží	50
7.4	Zpracování provizí	51

7.5	Webové standardy	53
7.6	Přístupný web	54
7.7	Ajax	54
8	Závěr	56
9	Reference	57
	Přílohy	57
A	Uživatelská příručka	58
A.1	Uživatelské role systému	58
A.2	Administrační rozhraní systému	58
B	Obsah přiloženého CD	65
C	Datový slovník	66

Seznam tabulek

1	Users	66
2	Distributors	66
3	Feeds	67
4	Floors	67
5	Categories	68
6	Category_Products	68
7	Export_Floors	68
8	Exports	69
9	Products	70
10	Manufacturers	70
11	Relateds	71
12	Photos	71
13	Parameters	71
14	Products_Parameters	72
15	Customers	73
16	Addresses	73
17	Countries	73
18	Orders	74
19	Taxes	74
20	Transports	75
21	Payments	75
22	Order_Addresses	75
23	Order_Items	76
24	Flags	76
25	Flag_Products	76
26	Delayed_Jobs	77
27	Group_Variants	77
28	Variants	77
29	Product_Variants	78

Seznam obrázků

1	Proces objednání zboží pomocí affiliate e-shopu	12
2	Prognóza růstu nákladů na affiliate marketing	13
3	Nejrozšířenější prosazované sektory	14
4	Affiliate síť Commission Junction	15
5	Affiliate síť TradeDoubler	15
6	Affiliate síť Zanox	16
7	Affiliate síť Potenza	18
8	Internetové obchody provozované společností VIVANTIS a.s.	19
9	Provize pro jednotlivé komodity společnosti VIVANTIS a.s.	20
10	Společnost INVIA.CZ nabízející partnerský program	21
11	Souhrn vybraných českých affiliate programů	21
12	Use Case – Systém	31
13	Use Case – Import zboží	33
14	Use Case – Export zboží do vyhledávačů	36
15	Use Case – Správa objednávky	38
16	Aktivitní diagram – Import zboží	39
17	Aktivitní diagram – Export zboží do vyhledávačů	41
18	Datový model	43
19	Třídní diagram	44
20	Sekvenční diagram – Import zboží	45
21	Architektura Model-View-Controller	47
22	Rozmístění souborů	49
23	Import kategorií pomocí transakce	51
24	Provize z objednávky	52
25	Průměrná denní provize za rok	53
26	Nástěnka s přehledem provizí	59
27	Formulář pro vložení nového uživatele	60
28	Výpis všech objednávek	60
29	Detail objednávky	61
30	Správa importů	62
31	Nastavení provize v detailu patra	63
32	Přidání exportu	63
33	Procesy	64

Seznam výpisů zdrojového kódu

1	Ukázkový program v jazyce Ruby	22
2	Ukázkový program 2 v jazyce Ruby	23
3	Ukázka použití metody find	24
4	Výpis všech článků seřazených sestupně podle datumu	24
5	Výpis všech článků Karla	24
6	Výpis všech článků Karla s titulkem World	24
7	Změna hodnoty atributu 1 v tabulce Article	24
8	Vztahy mezi tabulkami	25
9	Naplnění tabulek daty	25
10	Počet článků v kategorii Basketbal	25
11	Výpis kategorie do které patří článek Dokument	26

1 Úvod

Internet jako médium přináší nové a nové možnosti reklamy a usnadňuje sledování její efektivity jako žádné jiné. Jednou z forem online reklamy je Affiliate marketing, který je v Česku dosud hodně opomíjen a kterým se zabývá tato práce. Affiliate marketing je ale ve světě stále ve větší míře využívaný nástroj a toho si byli vědomi ve firmě Skvělý.CZ s.r.o., která patří již několik let mezi přední poskytovatele internetových řešení. Tato společnost se rozhodla vytvořit takovýto systém založený na Affiliate marketingu.

Tato práce se zabývá vývojem Affiliate B2C systému, který bude získávat veškeré zboží od dodavatelů a bude si vydělávat na provizích, které získá vlastník eshopu za každé prodané zboží, prostřednictvím jeho stránek.

K firmě Skvělý.CZ s.r.o. jsem se dostal prostřednictvím objektového jazyka Ruby a frameworku Ruby on Rails, ve kterém je tato práce implementovaná. Tato firma jako jedna z prvních v republice začala tento framework využívat a udělala o něm i jednu z přednášek na VŠB, které jsem se zúčastnil a hned jsem si jazyk Ruby a framework Ruby on Rails oblíbil. Proto jsem při výběru diplomové práce neváhal tuto firmu kontaktovat s výběrem tématu. Jak již bylo zmíněno, rozhodli jsme se vytvořit Affiliate B2C systém, tedy provizní systém, který bude postaven na technologii Ruby on Rails.

Každý takovýto systém, musí při svém procesu vývoje projít určitým cyklem, který se budu snažit popsat v následujících několika kapitolách.

Kapitola 2 – Affiliate marketing – seznamuje se základním principem affiliate marketingu.

Kapitola 3 – Použité technologie – popisuje jednotlivé technologie, použité v této práci. Hlavní částí této kapitoly bude představení objektového jazyka Ruby a frameworku Ruby on Rails, který je na jazyku Ruby postaven.

Kapitola 4 – Specifikace zadání – podrobně popisuje zadání naší práce, jednotlivé funkce systému, vstupy a výstupy.

Kapitola 5 – Specifikace požadavků – popisuje co má softwarový systém dělat pomocí specifikace jeho funkcionality. Požadavky jsou popsány ve formě use-case diagramu a aktivitního diagramu.

Kapitola 6 – Analýza a návrh – zkoumá specifikované požadavky z pohledu objektů, které lze nalézt v problémové doméně. Objekty popisuje pomocí sekvenčního a třídního diagramu. Model návrhu dále upřesňuje model analýzy ve světle skutečného implementačního prostředí.

Kapitola 7 – Implementace – ukazuje nezbytné vlastnosti systému a technologie použité při implementaci.

Kapitola 8 – Závěr – hodnotí výsledky mé práce a můj přínost.

Uživatelskou příručku pro návod jak zacházet se systémem najdeme v příloze A. Obsahuje také důležité přihlašovací údaje do systému.

2 Affiliate marketing

Internet byl stvořen jako médium, které bylo zaměřené především na informace. V poslední době se o něm začalo mluvit jako o výdělečné platformě. Internet nabízí nebývalé možnosti pro on-line reklamu. Umožňuje a hlavně usnadňuje sledování její účinnosti jako žádné jiné. V současnosti je velice oblíbenou, bohužel zatím v České republice trochu opomíjenou, součástí on-line reklamy affiliate marketing, který si představíme v tomto textu.

2.1 Úvod do affiliate marketingu

Jak jsme se zmínili v úvodním odstavci, affiliate marketing je určitá forma on-line reklamy. V Česku je také znám pod pojmem „provizní systém“, který vystihuje jeho podstatu. Obchodník pro propagaci své služby využívá stránek jiných serverů. Aby tyto servery odkazovaly podle obchodníkových představ, musí je obchodník nějakým způsobem motivovat. A jak to v dnešní době bývá, nejlepší motivací jsou peníze.

Affiliate tedy funguje na principu partnerství mezi stránkami prodejce výrobků nebo služeb (nazýváme ho **provozovatel affiliate programu**) a stránkami, které tyto produkty doporučují a odkazují na ně (tzv. **affiliate partner**). V podstatě jde o sdílení tržeb mezi provozovatelem affiliate a affiliate partnerem. Affiliate partneři dostávají finanční odměnu za přivedení návštěvníka, který na webových stránkách prodejce provede nějakou specifikovanou činnost. Tuto činnost můžeme chápat jako např. nákup výrobku nebo nějaké služby, registrace newsletteru apod.

Affiliate program si můžeme představit jako implementaci affiliate marketingu pro elektronický obchod, firemní prezentaci nebo jiné stránky.

Affiliate partner většinou využívá odkazů na stránky provozovatele affiliate. Tyto odkazy mohou být ve formě textu, obrázků jednotlivých produktů nebo dalších obrázků, které mají za cíl propagovat obchodníkovi výrobky potencionálním zákazníkům.

2.2 Počátky affiliate marketingu

Stejně jako optimalizace pro vyhledávače a marketing ve vyhledávačích, se také affiliate marketing dostává do českých zemí z USA. Máme sice zpoždění asi 4 až 8 let, ale zase přebíráme jen ty obchodní a marketingové modely, které jsou již dlouhodobě ověřeny.

Za prvního průkopníka v oblasti affiliate marketingu se považuje internetový obchod Amazon. Ten patří mezi nejstarší a největší na internetu. Amazon Associates je první affiliate program, který byl spuštěn v roce 1996.

Některé náznaky affiliate marketingu se objevili již před rokem 1996 na některých erotických webech. Amazon ale jako první dokázal přilákat obrovské množství provozovatelů webu a dokázal plně využít jejich potenciálu. S tímto „obrem“ na poli affiliate marketingu spolupracuje kolem milionu webů.

Affiliate sítě jsou portály, které nabízejí technologické řešení pro provoz a správu affiliate programů. Tyto sítě postupně začaly vznikat na konci devadesátých let minulého století. Nabízejí přístup k obrovskému množství provozovatelů webů, zajišťují programy

svým jménem, což je nesmírně důležité, protože to affiliate partnerům alespoň částečně zaručuje, že o dané peníze nemohou přijít.

2.3 Typy affiliate programů

Rozlišujeme dva typy affiliate programů, které označujeme:

1. **one-to-many** – nejrozšířenější affiliate programy, jsou veřejně otevřené pro velké množství affiliate partnerů. Velmi často bývají tyto programy k dispozici prostřednictvím affiliate sítí. Všichni provozovatelé webů mají naprosto stejné podmínky a ve většině případů se nevyžaduje exkluzivita. V praxi se tak může stát, že na 1 stránce můžete najít X navzájem si konkurujících affiliate odkazů (např. X prodejců letenek). Provozovatelé webu si sami vybírají, se kterými provozovateli affiliate (obchodníky) budou spolupracovat.
2. **one-to-one** – tyto affiliate programy jsou oproti prvnímu způsobu neveřejné, obchodníci si s affiliate partnery vyjednávají podmínky individuálně. Tento typ se vyskytuje u větších webů, které již mají známější produkty a produkty o které je zájem, a tudíž i lepší a silnější vyjednávací pozici směrem k majitelům (webmasterům) www stránek.

2.4 Provize pro affiliate partnery

Hlavním rozdílem mezi affiliate marketingem a internetovou reklamou je, že neplatíte za zobrazení reklamy, ale za skutečně realizované prodeje. Za různá náhodná kliknutí návštěvníků na reklamu, kteří o dané produkty vůbec nemají zájem tak nemusíte zbytečně platit peníze. Pro provozovatele affiliate programu je to velice efektivní nástroj.

Provize pro affiliate partnery může být vyplácena nejen z první transakce, kterou zprostředkovaný návštěvník uskuteční. Někteří provozovatele affiliate programu vyplácejí určité procento z každého nákupu, který daný návštěvník uskuteční.

Máme několik možností, jakým způsobem vydělat peníze přes affiliate marketing:

1. **Zaplacení za klik (Pay Per Click)** – zákazník se kliknutím na odkaz na stránkách affiliate partnera dostane na stránky obchodníka. Tento způsob odměny však nebývá běžnou praxí, protože to může přinášet zisk pouze affiliate partnerovi a zákazník na stránkách obchodníka nemusí strávit ani minutu.
2. **Zaplacení za prodej (Pay Per Sale)** – affiliate partner dostane zapláceno až za uskutečněný prodej nějakého výrobku nebo služby => získá určité procento z prodaného výrobku.
3. **Zaplacení za přivedení návštěvníka (Pay Per Lead)** – odměna je vyplácena za přivedení zákazníka na stránky obchodníka. Toto přivedení můžeme chápat jako např. registrace zákazníka na stránkách obchodníka. To je rozdíl oproti prvnímu způsobu, kdy jsme odměněni pouze za kliknutí na odkaz.

4. **Zaplacení za přivedení nových affiliate partnerů** – affiliate partner dostane zaplacení, že přivede nové affiliate partnery, kteří budou s obchodníkem spolupracovat.
5. **Zaplacení za zobrazení reklamy** – přiznání provize po pouhém zobrazení reklamy. Tento způsob v podstatě ošetřuje případy prvního způsobu, kdy zákazník může opsat adresu ručně do prohlížeče, aniž by musel klikat. Opět tento způsob není typickým příkladem affiliate marketingu. Obchodník ještě nemá vydělané peníze a přesto platí affiliate partnerovi.

Nejběžnější platební možnosti, spojené s affiliate marketingem, se tak stávají zaplacením za prodej a zaplacením za přivedení návštěvníka.

2.5 Přínos affiliate pro obchodníky

Jednou z největších výhod affiliate marketingu pro obchodníky jsou bezesporu náklady. Tyto náklady nejsou na rozdíl od běžné offline reklamy fixní, ale variabilní. To znamená, že náklady rostou jen tehdy, pokud zároveň roste i měřený výkon (prodeje, počet registrovaných zákazníků apod.). To přináší tu výhodu, že prostředky na reklamu jsou díky tomu vynaloženy mnohem efektivněji a tím se snižuje riziko neúspěchu kampaně.

Další velice podstatnou výhodou je, že tyto kampaně jsou plně automatizované. Tím se nám redukuje náklady spojené s jejich administrací a nákupem reklamy. Proto také může společnost inzerovat na takovém množství webu, které by jiným způsobem nebyla schopna oslovit, řídit a kontrolovat.

Reklama bývá na daných spolupracujících webech většinou přesně cílená, protože se provozovatel webu snaží co nejvíce na affiliate marketingu vydělat. Inzeruje tak na stránky výrobky a produkty se zaměřením na zájem svých návštěvníků. Jedná-li se např. o stránky, které jsou zaměřeny na sport a volný čas, webmaster se snaží zaměřit reklamu na tento okruh a nebude dávat reklamu např. na práci prášky apod. Pokud reklamu nebude přesně cílit na návštěvníky svého webu, reklama bude málo efektivní a tím přichází o možné výdělky z této reklamy.

Affiliate marketing má rovněž několik nevýhod pro obchodníky. Jednou z nevýhod je, že obchodník musí sdílet své zisky s další stranou. Další nevýhodou pro obchodníka je, pokud affiliate partner používá nevhodné prostředky pro prezentaci, které mají zákazníka přivést na jeho stránky, může obchodník přijít nejen o dobré jméno, které si vybudoval, ale také o ušlý zisk. Některé reklamy mohou být na stránkách affiliate partnera měsíce, než na ně někdo klikne a to třeba z důvodu špatně volitelného místa pro reklamu. Obchodník tak nemá kontrolu nad danou prezentací své obchodní značky. Některé affiliate programy tuto situaci řeší nastavením přesných podmínek. Při velkém počtu affiliate partnerů není ovšem reálné ohlídat vše. Někdy mohou provozovatelé webů konkurovat obchodníkům v nákupu kontextové reklamy a tím zvyšování ceny za klik. Proto si mnoho affiliate programů toto hlídá a zakazuje inzerovat na klíčová slova spojená se značkou prodejce.

2.6 Přínos affiliate pro provozovatele webů

Provozovatelům webu affiliate marketing přináší nové možnosti výdělku. To ocení zejména menší internetové projekty, které nemají možnost spolupracovat s mediálními agenturami a významnými inzerenty.

Největší nevýhodou affiliate marketingu pro provozovatele webu je přejímání rizika za kvalitu vytvořené kampaně. Může se stát, že affiliate partner udělal maximum pro zobrazení reklamy, ale už neovlivní obchodníkovou přeměnu návštěvníka v zákazníka. Z toho vyplývá, pokud obchodník nepřesvědčí zákazníka k nákupu, affiliate partner přichází o provize. To je jeden z důvodů, proč se affiliate marketing neosvědčil u velkých webových portálů, které nechtějí podstupovat výše zmíněné riziko. Tyto portály si mohou dovolit inzerovat jinými osvědčenými metodami.

Další poměrně velkou nevýhodou je, že není patřičná kontrola, zda obchodník affiliate partnerovi vyplácí provize za uskutečněné prodeje. Poslední viditelná nevýhoda je, pokud si návštěvník smaže cookies ze svého počítače, přichází affiliate partner o provizi. Informace ohledně cookies v této souvislosti jsou popsány v části „Affiliate marketing a eshopy“.

2.7 Affiliate marketing a e-shopy

V této části se zaměříme na fungování affiliate marketingu v oblasti internetového prodeje, konkrétně e-shopů. Jak jsme si v předcházející části řekli, máme několik způsobů vyplacení provize affiliate partnerovi. Tato část se zaměří pouze na jeden způsob placení provize a to je zaplacení za prodej. Nejobvyklejší model je, že pokud návštěvník provede na stránkách obchodníka nákup, affiliate partner dostane zaplacení 10 - 30 % z prodaného výrobku.

První způsob affiliate e-shopů je, že affiliate partner vloží odkaz s affiliate kódem na své stránky, diskuzních fór, e-mailů nebo přímo do kontextové reklamy ve vyhledávačích. Odměna se mu poté připíše na účet, pokud si návštěvník, který klikl na odkaz na stránkách affiliate partnera, na stránkách provozovatele affiliate (obchodníka) něco koupí.

Zadavatel vytvoří na svém webu speciální systém, který díky technologii cookies dokáže identifikovat, odkud zákazník přišel (od kterého affiliate partnera). V okamžiku uskutečnění první akce se z návštěvníka stává zákazník. Ověření probíhá tak, že systém se zeptá prohlížeče návštěvníka, zda má uloženou cookie systému. V cookie je zapsáno od kterého affiliate partnera návštěvník přichází a kdy to bylo. Žádný affiliate partnerem zprostředkovaný návštěvník nenakupuje ihned při první návštěvě stránek. Další návštěvu však již neprovede kliknutím na odkaz ze stránek affiliate partnera, ale půjde přímo. Díky cookies je toto možné a affiliate partner nepřichází o ušlý zisk. Doba, po kterou je cookie uložena se běžně pohybuje kolem 30 dní. Z toho vyplývá jedna velká nevýhoda o které jsme si již řekli v předcházejících částech. Pokud si zákazník smaže cookies a poté nakoupí, affiliate partner přijde o provizi.

Má diplomová práce a následně její implementace se bude zabývat druhým možným způsobem, jak vydělat na prodeji výrobků obchodníka. Tento způsob spočívá v tom, že

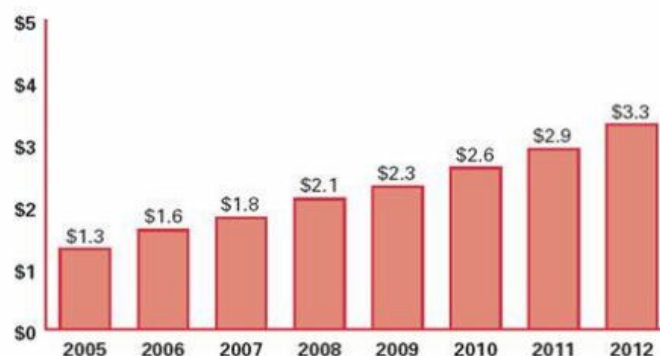


Obrázek 1: Proces objednání zboží pomocí affiliate e-shopu

affiliate partner má vlastní řešení internetového obchodu. Zákazník neprovádí nákup přes odkaz, umístěný na stránkách affiliate partnera, směřující na e-shop obchodníka, ale přímo nakupuje v internetovém obchodě affiliate partnera. Zákazník tak ani nepostřehne, že se stal součástí komunikace mezi obchodníkem a affiliate partnerem. Poskytovatel affiliate poskytne kompletní export dat, affiliate partner vytvoří e-shop a naimportuje kompletní data od provozovatele affiliate programu. Jakmile zákazník nakoupí vybrané zboží, je objednávka přeposlána k patřičnému dodavateli, který poté řeší dodání zboží zákazníkovi. O dodání zboží se již affiliate partner nezajímá, je to čistě věc obchodníka. Dodavatel poté zašle provizi z objednaného zboží affiliate partnerovi. Tato provize, jak jsme psali výše, činí většinou 10 – 30 % z prodaného zboží.

Na obrázku 1 je znázorněn proces nákupu zboží přes affiliate e-shop.

- Zákazník navštíví e-shop affiliate partnera se zájmem o koupi zboží
- Zákazník provede objednání vybraného zboží, jako v klasickém internetovém obchodě
- Vytvořená objednávka se přepoše poskytovateli affiliate programu
- Poskytovatel affiliate zpracuje objednávku a dodá objednané zboží zákazníkovi
- Poskytovatel affiliate zašle určité procento (10 – 30 %) affiliate partnerovi, od kterého objednávka přišla



Obrázek 2: Prognóza růstu nákladů na affiliate marketing

2.8 Affiliate marketing: situace v zahraničí

V následující části se podíváme, jak vypadá situace okolo affiliate marketingu v zahraničí a představíme si některý významné affiliate sítě.

Podle zprávy Econsultancy budou v roce 2009 ve Velké Británii tržby za affiliate marketing více než 4 miliardy liber. Oproti předchozímu roku se bude jednat o nárůst o 22 % z hodnoty 3,82 miliard liber. Zpráva rovněž říká, že 227 miliónů liber bude zaplacen na odměnách pro affiliate partnery.

V USA se rovněž výdaje na affiliate marketing neustále zvyšují a v roce 2008 vyplatili obchodníci svým affiliate partnerům a zprostředkujícím sítím provize ve výši 2,1 miliard dolarů. Odhaduje se, že v roce 2012 naroste hodnota tohoto čísla na 3,3 miliard dolarů. Na obrázku 2 vidíme prognózu růstu nákladů vydaných na affiliate marketing v USA podle průzkumu JupiterResearch (údaje jsou v miliardách dolarů).

Na obrázku 3 jsou zobrazeny procentuálně nejrozšířenější odvětví, které se prosadili v affiliate marketingu a celkově v e-shopech.

2.8.1 Zahraniční affiliate sítě

Zahraničních affiliate sítí je mnoho. Musíte se však ujistit, že pracujete s významnými společnostmi, které mají jasné nabídky affiliate programů, nabízí skutečnou podporu a v první řadě vám skutečně zaplatí.

Mezi největší zahraniční affiliate sítě patří: Commision Junction, TradeDoubler, Click-Bank, ShareaSale, ZanoX, LinkSahre, AffiliateFuture, buy.at a nebo Affiliate Window. V následující části se zblízka na některé vybrané sítě podíváme.

Commission Junction

Commission Junction je pravděpodobně nejlepší affiliate síť na světě, je to světový lídr v oblasti affiliate marketingu. Byla založena v roce 1998, sídlo má v Santa Barbaře, Californii a kanceláře po celém světě. V roce 2003 ji koupila společnost ValueClick. Kromě



Obrázek 3: Nejrozšířenější prosazované sektory

Spojených států působí v Německu, Španělsku, Francii, Švédsku a Velké Británii. Jejimi zákazníky jsou např. Apple Store, Dell, HP, Citibank nebo Travelocity. Síť nabízí přes 2300 affiliate programů, ale žádný z nich není přímo zaměřený na Českou republiku.

Nevýhodou této sítě je registrace. Oproti jiným sítím je složitější a vyžaduje zaslání daňového formuláře W-8BEN, který prokazuje, že webmaster je daňovým rezidentem mimo USA. Vyplnění tohoto formuláře také nepatří mezi triviální úkoly.

Výdělky se partnerům v České republice posílají každý měsíc, nejmenší částka, která se zasílá je 60 eur.

TradeDoubler

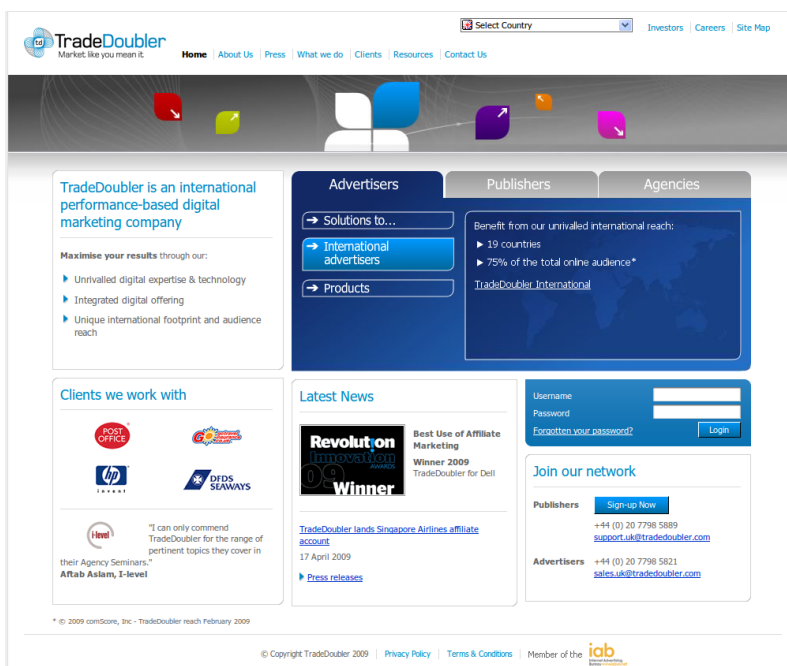
TradeDoubler je mezinárodní marketingová společnost založená ve Švédsku roku 1999. Dnes působí napříč Evropou a Japonskem, celkem působí v 19 zemích. Nabízí rozsáhlý výběr obchodníků, zahrnující velké značky typu American Express, Carphone Warehouse, HMV a mnoho dalších.

Počet prodejců, nabízející své výrobky a služby pomocí TradeDoubler se pohybuje přes 1 600. Tyto výrobce propaguje více než 128 000 affiliate partnerů. Mezi významné partnery, kteří s touto sítí spolupracují, patří AOL, eBay, Expedia, Sony nebo Yahoo.

Čeští webmasteri se mohou do této sítě registrovat, pokud na jejich stránky chodí návštěvníci ze země, kde tato síť působí. Jednotlivé národní verze affiliate programů nejsou propojené, proto se v každé zemi nachází různé affiliate programy. Na účet se webmasterům posílají peníze, pokud vyplacená částka překročí 30 liber.



Obrázek 4: Affiliate síť Commission Junction



Obrázek 5: Affiliate síť TradeDoubler

PUBLISHER | ADVERTISER | ZANOX | WIKI | BLOG

Select your language: Username LOGIN
 Register Forgotten your password?

zanox
we create partners

Join the top league of global online marketing

Check out our latest advertiser promotions here

Get the latest updates on our international programmes

AbeBooks_uk_EUR
Abebooks are offering an additional 2% on all sales when you achieve 5 or more transactions during May!!

AbeBooks_uk_GBP
New Abebooks API feed available now !!!

Biotherm UK
Discover a selection of sets at irresistible prices

Search Engine Marketing
Better landings due to better insights with zanox.
VIDEO Details

Global Alliance Partners
zanox 3rd party solution. Worldwide partnerships that really pay off.
VIDEO Details

Portals & Content Sites
High quality data increases sales.
VIDEO Details

E-mail & Incentivised Offers
Become successful on a global scale with e-mail marketing.
VIDEO Details

zanox news

WE ARE... ZANOX

Impressions of zanox GAS 2009

zanox Global Alliance Summit 2009

Warm-up now for the zanox Summer Party

International affiliate programmes and campaigns
Find the latest programme starts and top programmes in the zanox network
Details

Become a publisher with zanox
Join now for free!
Register now!

Top Affiliate Programmes

mypendo GameOn

amazon.co.uk

ecar insurance.co.uk

T-M-Jewin

Obrázek 6: Affiliate síť Zanox

Zanox

Firma Zanox byla založena roku 2 000 v Německu a postupně rozšiřuje svou činnost do dalších evropských zemí, USA a Číny, Austrálie. Zanox se tak stává jednou s největších firem v oboru. Zanox spolupracuje s více než 2 000 obchodníky a jeho affiliate programy inzeruje více než milion affiliate partnerů. Mezi významné partnery, patří Allianz, Amazon, Axa, Citibank, Deutsche Telekom, Telefonica O2, Vodafone, Orange, DADAmobile, Celldorado, Expedia, Jamster, Lycos, Procter & Gamble, Sixt, Staples, Neckermann, La Redoute a mnoho dalších.

Tuto síť jsem záměrně nechal na závěr, jelikož je pro české webmastery nejzajímavější ze všech zmiňovaných affiliate sítí. Nabízí několik programů, zaměřených na Českou republiku. Na Zanoxu se prezentují obchody jako Quelle, OTTO, bon prix, Beate Uhse, pneushop.cz, print24.cz a Schäfer Boutique, nebo letecké společnosti Air Berlin a Smart Wings.

Zanox rovněž nabízí několik německých a rakouských affiliate programů (e-shopy nebo cestovní kanceláře), které mohou čeští webmasteri využít.

Českým affiliate partnerům firma Zanox posílá výdělků na účet, pokud vyplacená částka přesáhne hodnotu 200 eur. Každý program je potom veden v určené měně a provize v různých měnách se vyplácí samostatně.

2.9 Affiliate marketing: situace na českém trhu

V předchozí části jsme se podívali na situaci ohledně affiliate marketingu v zahraničí, nyní se podíváme, jak to vypadá s affiliate marketingem na českém trhu.

Jak jsme z předcházející kapitoly zjistili, affiliate marketing je v zahraničí hodně populární a nabývá na významu. V České republice je situace opačná, stojí pořád na okraji zájmu.

Problém „českého“ affiliate marketingu začíná již v legislativě a vyžadované administrativě. Tento problém se týká finančních úřadů, které dosud nemají jasno, zda zařadit affiliate marketing jako součást podnikání a bude proto potřeba živnostenský list. Některým českým obchodníkům stačí pro spolupráci smlouva, zatímco někteří vyžadují živnostenský list.

Na Českém trhu velice chybí silný hráč, který by dokázal affiliate programy zaštitit. Podíváme-li se do zahraničí, tak většina affiliate programů se tam nabízí přes affiliate sítě. V Česku si naopak prodejci tyto programy vytvářejí a propagují sami. To způsobuje řadu komplikací, zejména složité dozvídání se o nových nabídkách, zdlouhavé čekání na dosažení minimální hranice pro vyplacení provizí a nemožnost kontroly, zda jsou vyplaceny odměny za skutečně všechny zprostředkované prodeje.

V Česku se affiliate marketing zatím rozšířil na několik málo odvětví, zejména cestování (prodej letenek a zájezdů) a omezeně v maloobchodě.

2.9.1 Affiliate síť Potenza

Potenza je jedinou affiliate sítí v České republice. Jejím vlastníkem je agentura Mather Advertures. Tato první affiliate síť vzbuzovala v době svého vzniku velké naděje, ale v současnosti nepřináší nic nového a dalo by se říci, že stagnuje. Potenza funguje nejen jako „reklamní tržiště“, jak sebe samou nazývá, ale nabízí rovněž reklamní kampaně, placené za klik.

Spolupráce s Potenzou se zahajuje vyplněním on-line formuláře a následným podepsáním písemné smlouvy. Affiliate partnerům se vyplácí provize od částky 1 000 Kč. Potenza využívá pro administraci kampaní řešení DirectTrack. Toto řešení ale není úplně dokonalé, nejenže není plně přeloženo do češtiny, což může některé partnery odradit, ale nesplňuje i technickou zpracovanost. Navíc Potenza, na rozdíl od zahraničních affiliate sítí, které se snaží procesy automatizovat, řeší mnoho věcí ručně (od fakturací po zpracování osobních údajů).

Potenza na svém webu nenabízí mnoho informací pro affiliate partnery. Chybí tam úplně možnost, prohlédnout si některé kampaně. Rovněž na tomto webu nenajdeme, kolik společnost účtuje obchodníkům jako svou provizi. Podle neoficiálních informací se mluví o 30 – 50 % z platby affiliate partnerům.



Obrázek 7: Affiliate síť Potenza

2.9.2 České affiliate programy

Pojďme se podívat na některé české affiliate programy, které jsou nabízeny společnostmi z různých odvětví. Mezi tyto programy patří například on-line cestovní agentura IN-VIA.CZ, internetový prodejce VIVANTIS, on-line cestovní agentura NetTravel, on-line prodejce elektroniky Pixmania, internetový obchod, zaměřený na digitální fotografie, kamery, dalekohledy Megapixel. Dalšími jsou například prodejci letenek Letuška nebo Královna.

Ve většině českých affiliate programech se platí buď za přivedení návštěvníka nebo za zprostředkování prodeje. Platbu za klik najdeme v těchto programech ojediněle. Provize jsou v Česku srovnatelné s platbami v zahraničím. Oproti zahraničí bývá velice častá kratší doba sledování zákazníka pomocí cookies. V zahraničí je doba platnosti většinou 30 dní, v Česku najdeme hodně programů, které mají tuto dobu kratší.

Jak jsme viděli na obrázku 3, nejrozšířenějším odvětvím byl prodej letenek a cestovní ruch. Nejinak je tomu i v Česku, kde jsou letenky nejlukrativnějším oborem, ale zároveň nejvíce konkurujícím.

Spoustu dalších českých affiliate programů naleznete na internetové adrese:
<http://webtrh.cz/16-ceske-affiliate-programy>

VIVANTIS a.s.

VIVANTIS a.s. je ryze česká společnost, která působí na internetovém trhu již od roku 2001. Jedná se o největšího internetového prodejce v ČR. Společnost se zabývá především prodejem parfémů, hodinek, šperků a produktů pro zdraví a krásu. Nabízí přes 100 000 produktů z 6 e-shopů.



Obrázek 8: Internetové obchody provozované společností VIVANTIS a.s.

Na svém webu <http://www.provizni-system.cz/> sdružuje affiliate program svých 6 e-shopů. Tyto e-shopy vidíme na obrázců 8.

Za každé prodané zboží může affiliate partner dostat provizi až do výše 10 % dle prodaného zboží. Provize z opakovaných objednávek rovněž činí 10 %. Tato provize se vyplácí již od částky 500 Kč a lze o ní žádat vždy maximálně 1x do měsíce. VIVANTIS nabízí možnost okamžité kontroly aktuálního stavu provizního účtu. Počet partnerů, spolupracujících se společností VIVANTIS a.s. se pohybuje kolem 2 000. Pokud zákazník navštíví stránky přes reklamní prvek a nenakoupí ihned, cookie, určená ke sledování zákazníka je platná 30 dní.

Společnost na provizích dosud vyplatila částku 5 386 871 Kč.

INVIA.CZ

Největší internetová cestovní agentura v České republice a autorizovaný prodejce zájezdů více než 200 renomovaných a pojištěných cestovních kanceláří působících na českém a slovenském trhu. Spolupracuje s největším on-line prodejcem zájezdů v České republice. Společnost vznikla v listopadu 2000 se sídlem v Praze. Nabízí na jednom místě neustále aktualizovanou nabídku katalogových i last minute zájezdů v přehledné, jednotné formě.

Společnost nabízí kromě internetu kamenná klientská místa ve více než 30 českých městech. Má své zastoupení dokonce v zahraničí Slovensku, v Polsku, Maďarsku a Rumunsku.

Partnerský program umožňuje propagovat webové stránky INVIA.CZ pomocí bannerů a vyhledávacích panelů, umístěných na stránkách affiliate partnera. V nabídce je rovněž, oproti jiným poskytovatelům affiliate, platba za klik. Pokaždé, když někdo klikne na banner, dostane affiliate partner 0,80 nebo 1,20 Kč (podle typu odkazu na jeho stránkách).

10% hodinky.cz	10% parfemy.cz	10% sperky.cz
10% krasa.cz	10% prozdravi.cz	10% nakupnicentrum.cz/hodinky
10% nakupnicentrum.cz/sexshop	10% nakupnicentrum.cz/oblecky-pro-psy	10% nakupnicentrum.cz/modni-doplňky
10% nakupnicentrum.cz/obleceni	10% nakupnicentrum.cz/spodni-pradlo	10% nakupnicentrum.cz/sperky-klenoty
10% nakupnicentrum.cz/zdravi	10% nakupnicentrum.cz/parfemy	10% nakupnicentrum.cz/kosmetika
5% nakupnicentrum.cz/hracky	5% nakupnicentrum.cz/nabytek	5% nakupnicentrum.cz/outdoor
3% nakupnicentrum.cz/naradi-zahrada	1% nakupnicentrum.cz/bile-zbozi	1% nakupnicentrum.cz/elektro
1% nakupnicentrum.cz/mobilni-telefony	1% nakupnicentrum.cz/pocitace	1% nakupnicentrum.cz/fotoaparaty

Obrázek 9: Provize pro jednotlivé komodity společnosti VIVANTIS a.s.

Společnost umožňuje mít více affiliate produktů pod jedním účtem (zájezdy, letenky, dárkový certifikát).

Za každou prodanou letenku obdrží affiliate partner 200 Kč nebo 1 % z ceny prodaného zájezdu. Může být také odměněn 2 % z ceny prodaného dárkového certifikátu. Firma vyplatí affiliate partnerovi odměnu ve výši 10 % z vyplacené provize (za prodaný zájezd nebo letenku) za každého jím přivedeného affiliate partnera. Podmínka je, že celková výše na jednoho přivedeného affiliate partnera nepřesáhne 20 000 Kč.

Minimální fakturovaná částka během jednoho kalendářního měsíce je 200 Kč, maximální činí 50 000 Kč. Cookie pro sledování zákazníka je platná 14 dní. Od roku 2006 společnost na provizích vyplatila částku 9 028 718 Kč.

2.9.3 Prognózy vývoje affiliate marketingu v ČR

Jak jsme si již řekli, situace na českém trhu ohledně affiliate marketingu zatím není na takové úrovni jako v zahraničí. Dá se ale předpokládat, že affiliate marketing bude v Česku pozvolna následovat zahraniční trendy a postupně se více rozšíří v oborech, jako jsou finance, on-line služby, zábava a maloobchod. Bude také záležet na tom, zda některá ze zahraničních affiliate sítí vstoupí na český trh.



Obrázek 10: Společnost INVIA.CZ nabízející partnerský program

Název programu	Provize	Platba za klik	Platnost uživatelské cookies	Počet partnerů	Vyplacené provize
Potenza	podle programu	podle programu	podle programu	300	
INVIA.CZ	1 % z ceny zájezdu, 200 Kč za prodanou letenku, až 20 000 za přivedení affiliate partnera	1 a 1,50 Kč	14 dní		9 028 718 Kč
VIVANTIS	10 % dle prodaného zboží	ne	30 dní	2000	5 386 871 Kč
NetTravel.cz	35% z prodaného zájezdu	ne	60 dní	817	
Pixmania	1 % z prodeje(1 – 60 700 Kč) 2 % (60 700 – 182 000 Kč) 3 % (více než 182 000 Kč)	ne	30 dní		
Aukro	150 Kč za nového uživatele, +30 % z jeho poplatků za rok	ne	14 dní		
Královna	obvykle 200 nebo 380 Kč za letenku	ne	nenabízí		
Letuška	250 Kč za letenku	ne	21 dní	1500	přes 10 000 000 Kč
Megapixel	až 5 % dle prodaného zboží	ne	45 dní		

Obrázek 11: Souhrn vybraných českých affiliate programů

3 Použité technologie

3.1 Ruby

Ruby je velmi mladý interpretovaný skriptovací jazyk, který je díky své jednoduché syntaxi snadný na naučení. Vychází ze základního principu: programování je **tvořivá činnost**, která vám má přinášet radost: [4]

„Věřím, že – alespoň do jisté míry – je smyslem života být šťastný. Na základě tohoto přesvědčení je Ruby navrženo tak, že je nejenom snadné, ale i zábavné v něm programovat. Ruby vám umožňuje soustředit se na kreativní stránku programování, a nepřidělová vám další starosti.“

–Yukihiro Matsumoto, předmluva k prvnímu vydání knihy [2]

Ruby je jazyk, který je plně objektově orientovaný a vychází z Perlu, Smalltalku, Eiffelu, Ady a Lispu. Oblast použití Ruby je, stejně jako dalších skriptovacích jazyků, především Pythonu a Perlu, poměrně široká. Nejčastěji se tyto jazyky používají při psaní různých skriptů, které nám usnadňují práci. Hodí se, díky své syntaxi a objektovosti, i k psaní větších projektů.

Počátky Ruby sahají do roku 1993, kdy se Japonec Yukihiro Matsumoto, velký zastánce objektově orientovaného programování a známý pod přezdívkou Matz, snažil najít nějaký skriptovací jazyk, který by mu plně vyhovoval. Perl mu nepřipadal dost výkonný a Python nebyl tolik objektově orientovaný, jak by si přál. Proto se rozhodl vytvořit vlastní jazyk. První verze byla uvedena v roce 1995 a Matz se rozhodl tento jazyk pojmenovat podle Rubínu - Ruby. Největší popularity se Ruby těší v zemi svého původu - Japonsku, ale v současnosti se už i dostává do dalších zemí.

Jak bylo zmíněno, Ruby má velice čistou syntaxi a snaží se přiblížit přirozenému jazyku (běžné mluvě). Díky tomu je kód dobře čitelný a snadný na psaní. Nejlépe si to můžeme představit ve výpisu 1

```
5.times { print "Ahoj!" }  
=> Ahoj!Ahoj!Ahoj!Ahoj!Ahoj!
```

Výpis 1: Ukázkový program v jazyce Ruby

Ačkoliv neznáme syntaxi jazyka, dokážeme na první pohled říct, co bude výsledkem programu, protože program přečteme jako běžnou řeč (v angličtině) *Five times print Hello* - 5 krát vytiskni Hello.

Tato „literární povaha“ je pro Ruby zásadní a hodně tímto chováním ovlivnila framework Ruby on Rails.

Ruby vám umožňuje se skvěle vyjádřit, a nemusíte u toho psát tucty řádků komentářů — kód v Ruby je srozumitelný sám od sebe. [4]

Výsledkem je, že i neprogramátoři v Ruby mohou bez problému kód ve výpisu 2 přečíst.

```
print ["banan", "citron", "ananas"].sort.last.capitalize
=> Citron
```

Výpis 2: Ukázkový program 2 v jazyce Ruby

Programování v Ruby se zkrátka stalo zábavou a programujeme, protože nás to baví. Podrobnější informace o tomto jazyku se dozvíte v literatuře [2]. Velkou popularitu si Ruby získalo díky frameworku Ruby on Rails, se kterým se seznámíme v následující části.

3.2 Ruby on Rails

Ruby on Rails je revoluční programovací nástroj, jenž urychluje a zjednodušuje tvorbu webových stránek a jejich aplikací. Díky zapojení přímočarého jazyka Ruby a vývojové platformy, kterou poskytují Rails, můžeme ve velmi krátkém čase vytvořit plnohodnotné webové aplikace, i když člověk není zkušený programátor.[3]

Mnoho aplikací mají hodně kódu, který se neustále opakuje. Ať už jsou to nějaké výpisy např. detail objednávky pro zákazníka a detail objednávky pro systém nebo formuláře, které jsou pořád stejné (např. Přidání a editace tabulky). Ruby on Rails si proto zakládá na myšlence **DRY - Don't Repeat Yourself – Neopakuj sám sebe**, tzn. že každý kousek kódu je v systému právě na jednom místě. Tak vznikají různé pomocné metody a různě částečné pohledy (partial views), které napíšeme jen jednou a poté se v jiném pohledu na ně odkazujeme. To dělá Rails aplikaci přehlednou a nenutí nás psát pořád dokola, co již jednou máme napsané. [1]

Konvence má přednost před konfigurací – je klíčovým principem každé Ruby on Rails aplikace. O co jde? Mnoho aplikací je si hodně podobných. Načtou data z databáze, transformují do nějaké datové struktury jako jsou pole, hashe a objekty a poté s nimi provádějí různé manipulace, které následně vypisují do HTML. To samé platí pro ukládání dat do databáze. Pomocí formulářů se nasbírají informace, provede se s nimi nějaká manipulace a výsledek se uloží do databáze. Problém ovšem je, že málokomu se chce začínat psát aplikaci, jak tomu je v PHP, od `mysql_connect` a `SELECT * FROM Table ORDER BY nazev`. Proto je v Railsech pravidlo: **Autor konfiguruje pouze to, co se liší od běžného nastavení**. Dobrým příkladem je, máme-li model Article, aplikace automaticky hledá data v tabulce Articles, tj v tabulce, která je množným číslem modelu (Person – People apod.). Samozřejmě, pokud chceme mít model Article a přistupovat k tabulce Clanky, da se to pochopitelně nastavit jedním příkazem.

Ruby on Rails aplikace je rozdělena na vrstvy, kde každá má svůj specifický úkol. Tyto vrstvy nazýváme Model-View-Controller. Co je jejím úkolem si popíšeme později. Nyní nám stačí vědět že aplikace má 3 vrstvy a každá má nějaký úkol. My si nyní ukážeme

příklady na vrstvu model, která má na starosti manipulace s daty – je to ta část systému, kterou jsme si popisovali v předchozím odstavci, že model Article je napojený na tabulku Articles.

ActiveRecord je srdcem každé Rails aplikace a dokonale demonstrovuje principy, které jsme si řekli (Konvence má přednost před konfigurací). ActiveRecord můžeme nazvat jako návrhový vzor, který mapuje databázové tabulky na jednotlivé třídy, řádky mapuje na objekty (instance tříd) a sloupce na atributy. To nám ušetří psaní onoho `SELECT * FROM`, čímž nám zefektivní práci na daleko vyšší úrovni. Proto místo kódu `SELECT * FROM Articles WHERE id = 1` nám stačí napsat přehledné, výstižné a krátké

```
Article . find (1)
```

Výpis 3: Ukázka použití metody find

kde Article představuje třídu, která se mapuje na tabulku Articles a metoda find vyhledá položku s id 1.

Následující příklady ukazují různé varianty metody find.

```
Article . find (: all , :order => "created_at_DESC")
```

Výpis 4: Výpis všech článků seřazených sestupně podle datumu

```
Article . find_all_by_ autor ("Karel")
```

Výpis 5: Výpis všech článků Karla

```
Article . find_all_by_ autor_and_ titulek ("Karel", "World")
```

Výpis 6: Výpis všech článků Karla s titulkem World

Za metodou find můžeme dávat různé atributy z tabulky, viz předchozí příklad, čímž bude program čitelnější oproti příkazům, které bychom museli udělat se Selecty.

Jak jsme si řekli, Rails aplikace umožňují nejen vypisovat data, ale také umožňují manipulovat s daty. Příklad 7 ukazuje, jakým způsobem udělat, pokud chceme změnit obsah položky v tabulce Article.

```
article = Article . find (1)
article . autor = "Karel"
article . save
```

Výpis 7: Změna hodnoty atributu 1 v tabulce Article

Opět je zápis daleko přehlednější oproti klasickému sql.

Validace vstupních dat

Velice mocným nástrojem je validace vstupních dat, který nám kontroluje správnost dat ve formulářích. Chceme-li např. aby ve formuláři pole název nesmělo být prázdné uděláme to příkazem *validates_presence_of :nazev*

Pokud bychom takovýto prázdný článek chtěli uložit, dostaneme chybovou hlášku. Existuje spousta metod na kontrolu validací, jako kontrola, zda je obsah číslo, zda je správná délka, správný formát, zda se již prvek neopakuje v tabulce apod.

Vztahy mezi objekty

Jak víme z databází, mezi jednotlivými tabulkami existují různé vztahy, typu 1:1, 1:M a M:N. Tyto vztahy můžeme jednoduše zapsat do modelu, který je spojený s tabulkou. Máme-li tabulku Articles a tabulku Categories, tak vztahy mezi tabulkami jsou 1 článek má 1 kategorii, ale 1 kategorie má mnoho článku, proto je vztah 1:M. V naší aplikaci to bude vypadat následovně.

```
class Article < ActiveRecord::Base
  belongs_to :category
end
class Category < ActiveRecord::Base
  has_many :articles
end
```

Výpis 8: Vztahy mezi tabulkami

Pro metodu *belongs_to* máme pomůcku, že pokud tabulka obsahuje cizí klíč, odkazující na jinou tabulku, dáváme pokaždé *belongs_to*. Pokud máme vztah 1:1, dáme u jedné tabulky metodu *has_one*. Vztah typu M:N řešíme pomocnou vazební tabulkou.

Na následujícím příkladě 9 si ukážeme naplnění tabulek daty

```
article = Article.new
article.nazev = "Dokument"
article.obsah = "Obsah dokumentu"
article.save
category = Category.new(title => "Basketbal")
category.articles << article
category.save
```

Výpis 9: Naplnění tabulek daty

Posledním výpisem přiřadíme vybrané kategorii článek
Následující ukázky jsou jen pro zajímavost.

```
category = Category.find_by_nazev("Basketbal")
print category.articles.count
```

Výpis 10: Počet článků v kategorii Basketbal

```
article = Article.find_by_nazev("Dokument")
print article.category
```

Výpis 11: Výpis kategorie do které patří článek Dokument

Vidíme, že vztahy mezi tabulkami lze řešit jednoduchým a přehledným způsobem.

Migrace

Poslední věcí, kterou bych chtěl zde zmínit jsou migrace. Migrace slouží pro vytváření, mazání editování databázových tabulek a umožňuje tyto tabulky naplňovat daty. Velkou předností migrací je, že zápis pro vytváření tabulek je nezávislý na databázi, tzn. že bude fungovat pro všechny běžně dostupné databáze (MySQL, Oracle atd.). Nemusíme tak řešit problém, že náš poskytovatel přejde na jiný typ databáze apod. Hlavní výhodou migrací je možnost se vracet v čase, podobně jak to funguje u správy verzí (svn, cvs). To nám dává obrovský nástroj, pokud chceme přidat jeden sloupeček do tabulky, přejmenovat sloupec nebo smazat určitou tabulku, nemusíme mazat vše a nepřijdeme tak o naše data.

Pluginy

Ruby on Rails obsahuje spoustu nástrojů, které ušetří vývojáři práci. Vývojář se tak nemusí zabývat částmi kódu, které již někdo napsal a poskytl je ve formě pluginu a věnuje se jen konkrétním věcem ve své práci. Mezi takovéto pluginy patří např. `login_generator`, díky kterému nemusíme psát kompletní přihlašovací systém, jen si necháme celý kód vygenerovat a poté si jen upravuje vzhled a přesměrování, když nám nevyhovuje. Dalším velice užitečným pluginem `attachment_fu`, díky němuž můžeme jednoduše nahrávat soubory na server. Rovněž umožňuje nastavit, na jakou velikost chceme obrázky rozdělit, čímž dostaneme v databázi např. tři různé rozměrově velké obrázky. Poslední velmi užitečným pluginem, který se mi hodil, je `acts_as_list` který zařídí, aby se vypisovali položky seznamu v daném pořadí, pomocí zadané pozice a navíc dokáže uživateli poskytnout nástroj, pomocí něhož si může uživatel měnit sám pořadí těchto položek.

To bylo mírné seznámení s Ruby on Rails, pro podrobnější informace velice doporučuji publikaci [1], pokud neovládáte příliš anglický jazyk, vynikající blog na webu [4].

3.3 MySQL

Jako databázový systém jsem zvolil MySQL. Jedná se o databázový systém, který byl vytvořen švédskou firmou MySQL AB a nyní vlastněný firmou Sun Microsystems, která spadá pod společnost Oracle. Má dvojí licencování, jednak bezplatnou GPL a jednak komerční placenou licenci. Jedná se o multiplatformní relační databázi, se kterou se komunikuje prostřednictvím jazyka SQL. Podporuje nejrozšířenější programovací jazyky jak C, C++, C#, Java, PHP, Ruby on Rails aj. V současné době se MySQL používá hlavně pro tvorbu webových aplikací. [7]

3.4 Javascript

Je multiplatformní objektově orientovaný skriptovací jazyk, který se používá na internetových stránkách. Jeho syntaxe je podobná programovacím jazykům C, C++ nebo Java. Javascript je klientský skript, tzn. že se program odesílá se stránkou klienta do prohlížeče a teprve tam je vykonán, na rozdíl od ostatních jiných interpretovaných programovacích jazyků (např. PHP a ASP), které se spouštějí na straně serveru ještě před stažením z Internetu. Z toho plynou jistá bezpečnostní omezení, JavaScript např. nemůže pracovat se soubory, aby tím neohrozil soukromí uživatele.

Autorem je Brendan Eich z tehdejší společnosti Netscape. Ten jazyk nejdříve pojmenoval jako Ecma, poté název změnil na LiveScript a nakonec byl pojmenován to dnešní podoby jako Javascript.

Program v JavaScriptu se obvykle spouští až po stažení WWW stránky z Internetu (tzv. na straně klienta). [8]

3.5 AJAX

Zkratka znamená Asynchronous JavaScript and XML. Obecně se tak označují technologie vývoje interaktivních webových aplikací, které mění svůj obsah bez nutnosti jeho znovu-načítání. Při změně nějaké prvku na stránce se tak nemusí překreslovat kompletní celá internetová stránka, ale změní se jen obsah daného prvku. Na rozdíl od klasických webových aplikací poskytují aplikace, u nichž je použita technologie AJAX, příjemnější uživatelské prostředí, kdy se aplikace může chovat jako klasická desktopová aplikace. Tato technologie vyžaduje použití nejmodernějších internetových prohlížečů.

Termín Ajax se poprvé objevil v dubnu 2005. Velká popularita a rozšíření AJAX začala službami společnosti Google. Jednou z nevýhod této technologie možnost zvýšení http požadavků. [9]

3.6 XHTML

Zkratka znamená eXtensible Hypertext Markup Language (rozšiřitelný hypertextový značkovací jazyk). Jedná se o značkovací jazyk pro tvorbu hypertextových dokumentů v prostředí WWW. Je vyvíjen společností W3C. Jedná se o novější normu jazyka HTML, protože se jazyk HTML dlouho nevyvíjel a zůstal na verzi 4.0.1. V současnosti se vytváří nová verze HTML 5 a její XML varianta XHTML 5. XHTML oproti HTML nepřináší žádné možnosti navíc, spíše omezení, z hlediska validity výsledného dokumentu. [10]

3.7 CSS

Zkratka CSS znamená Cascading Style Sheets – Kaskádové styly. Popisují způsob zobrazení stránek napsaných v jazycích HTML, XHTML, nebo XML. Kaskádové se jim říká, protože umožňují na sebe vrstvit definice stylů. Byl navržen, stejně jako HTML a XHTML organizací W3C. Hlavním smyslem toho jazyka je umožnit návrhářům oddělit vzhled dokumentu od jeho struktury a obsahu. [11]

4 Specifikace zadání

Úkolem této práce jen navrhnout a poté realizovat jádro Affiliate B2C systému v technologii Ruby on Rails. Hlavním cílem této práce bude vytvoření systému, který poskytuje provozovateli komplexní affiliate podporu pro neomezený počet dodavatelů. Nezbytnou součástí práce budou patřičné importy od dodavatelů, exporty produktů do vyhledávačů, automatické přepočty cen, obrátů a zisků jednotlivých komodit. Samozřejmostí bude zaměření na účetní podporu pro provozovatele. Systém bude rovněž splňovat co nejlepší uživatelskou přívětivost pomocí moderních technologií AJAX, Web 2.0.

Prostřednictvím administračního rozhraní systému(backend) bude uživatel moci spravovat kompletní e-shop. Může spravovat jednotlivá patra e-shopu, které v podstatě slouží jako patra(oddělení) v klasické kamenném obchodě např. hodinky, šperky, kosmetika, mobilní telefony apod. V každém patře si bude moci uživatel nastavit, zda je povoleno nebo ne, výši provize, jakou bude mít z každého prodaného výrobku. Patra a distributoři budou v systému uloženy napevno, tzn. administrátor nebude moci přidávat nové hodnoty, jen měnit stávající dodatečné údaje. Je to z důvodu, že systém bude schopen pracovat jen s určitým počtem dodavatelů, které právě podporuje. To nám usnadní práci s importy, kdy má každý dodavatel rozdílný tvar XML souboru. Pokud přibude nové patro nebo distributor, bude se to nahrávat přes Rails úlohu Rake tasks.

Stěžejním bodem bude práce s importy. Tyto importy se nastavují v daném patře. Umožněno bude povolení stahování(povolení patra), nastavení přihlašovacích údajů k danému feedu. Samotný importu bude probíhat dvojím možným způsobem. Ručně, stisknutím příslušného tlačítka pro import a veškerá práce bude probíhat na pozadí, takže se systémem se bude moci dále pracovat. Jednotlivé procesy budou zobrazeny na příslušné stránce procesů pro přehled, co se právě děje, které patro se importuje a které čeká na import. Druhá možnost je automaticky, nastavením času kdy spouštět úlohy, přes CRON.

Systém bude rovněž schopen pracovat s exporty zboží do vybraných vyhledávačů. Podobně jako importy bude mít dvě možnosti spuštění a to jak ručně tak automaticky. Poté se vygeneruje patřičný xml soubor na dané místo na serveru. Uživatel si adresu zkopíruje a může zaregistrovat do vyhledávače.

U objednávek budeme moci nastavovat stav objednávky, zda je nová, v procesu vyřizuje se, stornována nebo vyřízená. Pokud je objednávka vyřízená a v daném patře je nastavena výše provize, zobrazí se u objednávky provize z této objednávky. Objednávky, které jsou mladší než 14 dní se budou po stisknutí tlačítka exportovat do xml souboru pro ekonomický a informační systém Pohoda.

Nepostradatelnou věcí bude správa uživatelů, pro přístup do systému, zobrazení jednotlivých importovaných produktů, správa dopravy, platby a daně.

Jednou z nejdůležitějších věcí bude zobrazení hodnoty jednotlivých provizí za objednávku, celkové sumy provize za měsíc a rok, průměrné denní a roční provize. Mezi jednotlivými roky se bude dat přepínat. Provize budou také zobrazeny v přehledném grafu.

4.1 Vstupy

- Distributoři
- Importy
- Exporty
- Patra
- Produkty
- Výrobci
- Kategorie pro jednotlivé produkty
- Zákazníci
- Objednávky
- Dopravy
- Platby
- Daně
- Obrázky pro jednotlivé produkty

4.2 Výstupy

- Zobrazení produktů a jejich detailních informací
- Zobrazení objednávek a jejich detailů
- Zobrazení dopravy a platby
- Zobrazení daní
- Zobrazení hodnoty jednotlivých provizí(za objednávku, suma za měsíc, rok, průměrné denní a roční provize)
- Zobrazení pater
- Zobrazení jednotlivých exportů
- Zobrazení stavu procesů při importu nebo exportu zboží

5 Specifikace požadavků

Cílem specifikace požadavků je popsat co má softwarový systém dělat prostřednictvím specifikace jeho funkcionality. Modely specifikace požadavků slouží k odsouhlasení zadání mezi vývojovým týmem a zadavatelem. [5]

Nejdříve popíšu specifikaci požadavků, pomocí případů užití a poté pomocí aktivitních diagramů ukážu tok činností na vybraných částech systému.

5.1 Specifikace pomocí případů užití

Ke specifikaci případů užití využiji jazyk UML a jeho diagramy případů užití (Use Case). Tyto diagramy nám definují vztahy mezi případy užití systému a aktéry stojící vně systému. Případy užití specifikující vzory chování realizované softwarovým systémem. Každý případ užití lze chápat jako posloupnost vzájemně navazujících transakcí vykonaných v dialogu mezi aktérem a vlastním softwarovým systémem. Jako aktéry pak definujeme uživatele či jiné systémy, kteří budou vstupovat do interakce s vyvíjeným softwarovým systémem. [6]

Nejdříve ukážu pomocí Use Case diagramů celý systém – jeden velký Use Case, který poté budu rozebírat do menších celků.

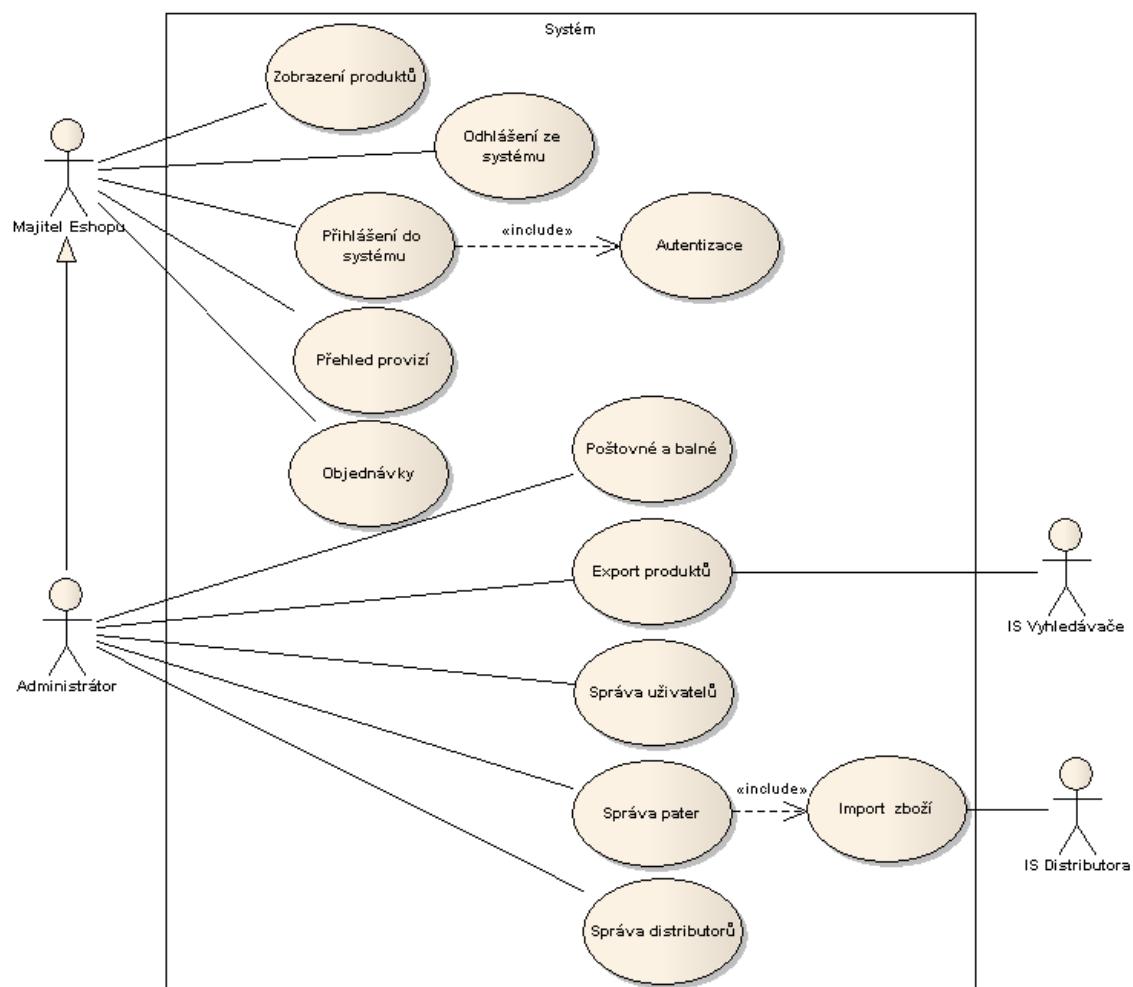
5.1.1 Systém

Diagram na obrázku 12 je Use Case diagram celého systému. Systém se skládá jen z administrativní části(backend), ze které se ovládá celé jádro affiliate internetového obchodu.

Slouží ke správě celého systému, ať už se jedná o správu importu a exportu zboží, správu objednávek, správu uživatelů backend části, správu dopravy, platby a daně. Spravovat se budou rovněž dodavatele a jednotlivá patra, ale pouze update daného záznamu. Nebude dána možnost vkládat distributory, importy(patru) a jednotlivé feedy přes toto rozhraní. V neposlední řadě zde bude zobrazena výše provizí, které dostaneme za každou vyřízenou objednávku.

Uvnitř systému rozlišujeme 4 základní typy aktérů:

- **Majitel eshopu** – může prohlížet stav provizí, zobrazovat jednotlivé produkty eshopu, prohlížet detaily objednávky a nastavovat jim stav.
- **Administrátor** – má veškerá práva a je správcem celého systému.
- **IS vyhledávače** – vzdálený systém u konkrétního vyhledávače, který si stahuje naše vytvořené feedy pomocí exportů.
- **IS distributora** – vzdálený systém u distributora, který nám poskytuje produkty ve formě xml souborů.



Obrázek 12: Use Case – Systém

5.1.2 Import zboží

Diagram na obrázku 13 zobrazuje Use Case diagram importu zboží. Importu ze zúčastní dva aktéři. Administrátor, který správu pater a tudíž i importů ovládá ve webovém rozhraní a vzdálený IS distributora, od kterého stahujeme xml feed soubory.

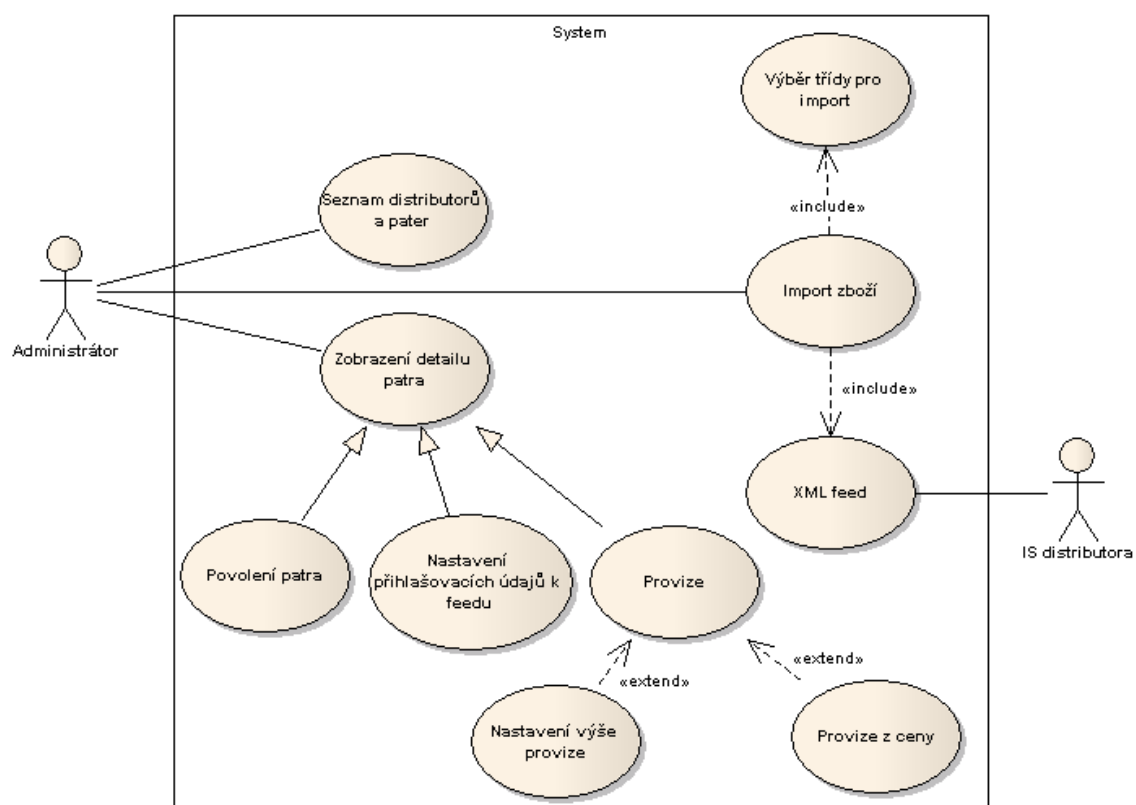
Administrátor může jen upravovat hodnoty daného patra, kromě názvu. Nemůže tak patro smazat nebo přidávat nové. Je to z důvodu spolehlivého parsování xml souboru, pokud přijde nový dodavatel s novým patrem. XML feed by nemusel být kompatibilní s existujícími knihovnamy pro zpracování parsování tohoto xml souboru. Patra se tak stejně jako distributoři vkládají do systému ručně např. Rails tasks, ne přes web rozhraní.

V detailu patra se nastavují hodnoty jako popisek patra, povolení patra, které slouží k zobrazení patra v eshopu a zároveň pro povolení importu zboží do tohoto patra. Stežejním bodem je zadání hodnot výše provize.

Provize může být zadávána v procentech u každého patra a její výše vypočítávána několika způsoby:

1. **Navýšení provize** – cena nabízeného produktu se zvýší o procentuální hodnotu, danou v nastavení patra, importovaného produktu. Výsledná provize z objednávky se spočítá jako rozdíl nové ceny produktu a původní ceny, která byla v xml souboru importovaného produktu. Např. cena produktu v xml souboru bude 1000 Kč, patro bude mít nastavenou provizi 10 % z prodaného výrobku, cenu produktu, který importuji do svého shopu zvýším o těch 10 %, takže cena produktu v mém shopu bude 1100 Kč. A výsledná provize bude rozdíl nové a původní ceny, tedy 100 Kč.
2. **Provize z ceny** – cena produktu bude stejná jako v xml feedu, provize se přímo spočítá. Např. cena produktu 1000 Kč a provize, která bude např. 10 %, bude 100 Kč na této objednávce.
3. **Kombinace** – Kombinace obou předchozích způsobů. Jednak navýším cenu produktu a provizi získám odečtením, a jednak provizi spočítám klasicky. Např. 5 % + 5 %. Produkt má cenu 1000 Kč. Cenu produktu podle prvního způsobu zvýším o 5 % na 1050 Kč. Provize bude 1050 – 1000 = 50 Kč. Druhým způsobem zjistím provizi ze zbývajících 5 %, která dělá 50 Kč. Výsledná provize je součet obou hodnot => 100 Kč.

Poslední možností je u patra nastavení přihlašovacích údajů, které mohou někteří distributoři vyžadovat k přístupu k xml souboru. Pokud nyní uživatel stiskne tlačítko „Importovat“, dojde k zařazení úlohy importu do fronty úloh. Pokud je úloha první, dojde okamžitě ke spuštění importu, stahováním dat z xml souboru, umístěném na vzdáleném IS distributora. Úloha se spustí na pozadí, což nám umožňuje pracovat dále se systémem a neblokuje se nám tak přístup na stránky. Pokud nějaká úloha běží, proces čeká na dokončení předchozí úlohy.



Obrázek 13: Use Case – Import zboží

Hlavní scénář:

1. Uživatel se rozhodne stáhnout produkty pro vybrané patro
2. Uživatel vybere patřičné patro stisknutím tlačítka
3. Systém zobrazí formulář s detailem patra(název, popisky, povolení patra, provize, uživatelské přihlašovací údaje pro přístup k feedu)
4. Uživatel zatrhne povolení patra
5. Uživatel nastaví hodnotu provize v inputu navýšení provize(cena produktu se zvýší o dané procento)
6. Systém zobrazí celkovou hodnotu provize v procentech jako součet obou druhů provizí
7. Uživatel stiskne tlačítko uložit patro
8. Systém zkontroluje správnost údajů
9. Údaje jsou v pořádku, systém je uloží do systému
10. Systém odešle informační zprávu o úspěchu uložení uživateli
11. Systém zobrazí výpis všech dodavatelů a jejich pater
12. Uživatel stiskne tlačítko „Importovat“
13. Systém zkontroluje, zda je patro povoleno
14. Patro je povoleno
15. Systém zjistí jakou třídu pro import má použít pro zpracování xml souboru
16. Systém přenesse úlohu do fronty úloh na pozadí, aby se dal web dále ovládat
17. Systém zjistí, že se jedná o první úlohu a nastaví jí stav běžící
18. Systém spustí import zboží vybraného patra z xml souboru ze stránek distributora
19. Systém zjistí, že hodnota navýšení provize je větší než 0, zvýší se cena produktu z xml souboru
20. Systém úspěšně dokončí import patra
21. Systém smaže danou úlohu z fronty úloh
22. Systém odešle informační zprávu o úspěchu uživateli

Rozšíření:

3a – Systém načte formulář

3a1 – Systém načte uložené údaje do formuláře

3a2 – Uživatel upraví potřebné hodnoty, skok na bod 7

5a – Uživatel chce vyplnit jiný typ provize

5a1 – Uživatel chce vyplnit provizi z ceny

5a1.1 – Uživatel vyplní údaj pro provizi z ceny, skok na bod 6

5a2 – Uživatel chce vyplnit oba údaje

5a2.1 – Uživatel vyplní hodnoty do obou typů provizí, skok na bod 6

7a – Údaje nejsou v pořádku

7a1 – Systém nahlásí uživateli chybu o nesprávném formátu nebo nevyplnění potřebných údajů, skok na bod 3

14a – Patro není povoleno

14a1 – Systém informuje uživatele o neoprávněné operaci, skok na bod 11

17a – Úloha není první ve frontě

17a1 – Systém nastaví této úloze stav čekající, skok na 14

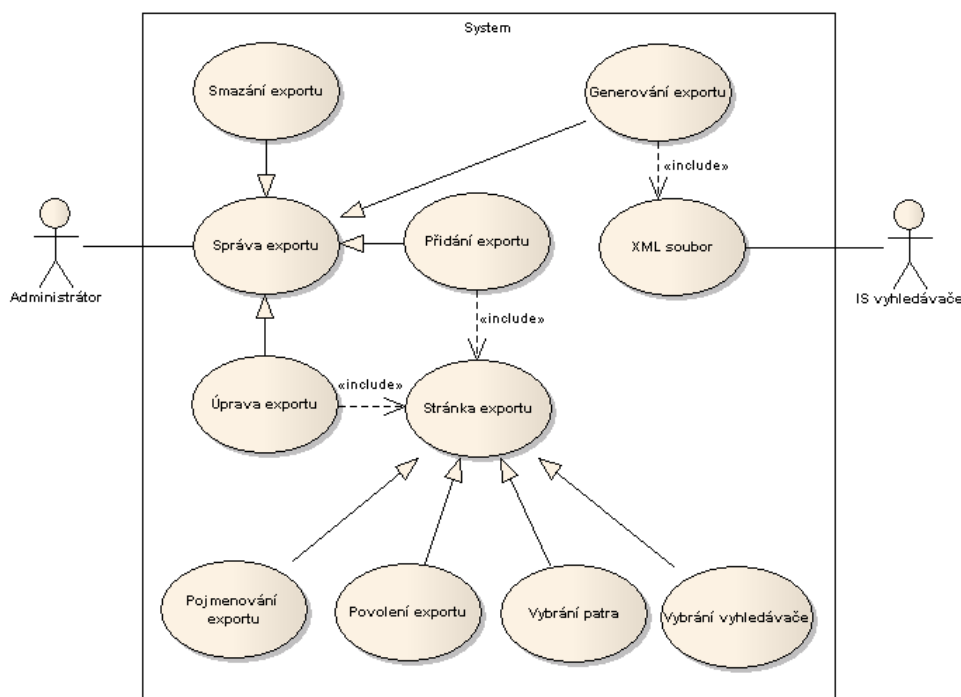
19a – Hodnota provize není nastavena

19a1 – Systém nezvýší cenu produktu, skok na 19

5.1.3 Export zboží

Diagram na obrázku 14 zobrazuje Use Case diagram exportu zboží do vyhledávačů. S exportem přijdou do styku dva aktéři. Administrátor, který správu exportu ovládá ve webovém rozhraní a vzdálený IS vyhledávače, který si pravidelně stahuje náš vygenerovaný xml soubor.

Administrátor může exporty, mazat, upravovat a přidávat. Přidání nebo úprava zahrnuje pojmenování exportu, vybrání vyhledávače, pro který se bude exportovat zboží, povolení exportu a vybrání patra, ze kterého se budou produkty exportovat. Pokud uživatel stiskne tlačítko exportovat na hlavní stránce, dojde k zařazení úlohy exportu do fronty úloh. Pokud je úloha první, dojde okamžitě ke spuštění exportu zboží do xml souboru. Vygenerovaný XML soubor si poté v pravidelných časových intervalech stahuje vzdálený IS vyhledávače. Úloha se stejně jako u importu spustí na pozadí. Pokud nějaká úloha běží, proces čeká na dokončení předchozí úlohy.



Obrázek 14: Use Case – Export zboží do vyhledávačů

Hlavní scénář:

1. Uživatel chce do systému vložit nový export
2. Uživatel stiskne tlačítko „Přidat nový export“
3. Systém zobrazí formulář pro nastavení exportu (název exportu, název xml souboru, povolení exportu, výběr vyhledávače pro export, seznam pater)
4. Uživatel vyplní formulář, nastaví vyhledávač, pro který se bude exportovat, vybere patro pro které se bude exportovat zboží
5. Uživatel stiskne tlačítko pro uložení údajů
6. Systém zkontroluje správnost údajů
7. Údaje v pořádku, systém uloží údaje do systému
8. Systém odešle informační zprávu o úspěchu uložení uživateli
9. Systém zobrazí výpis všech exportů
10. Uživatel stiskne tlačítko „Exportovat“
11. Systém zkontroluje, zda je export povolen

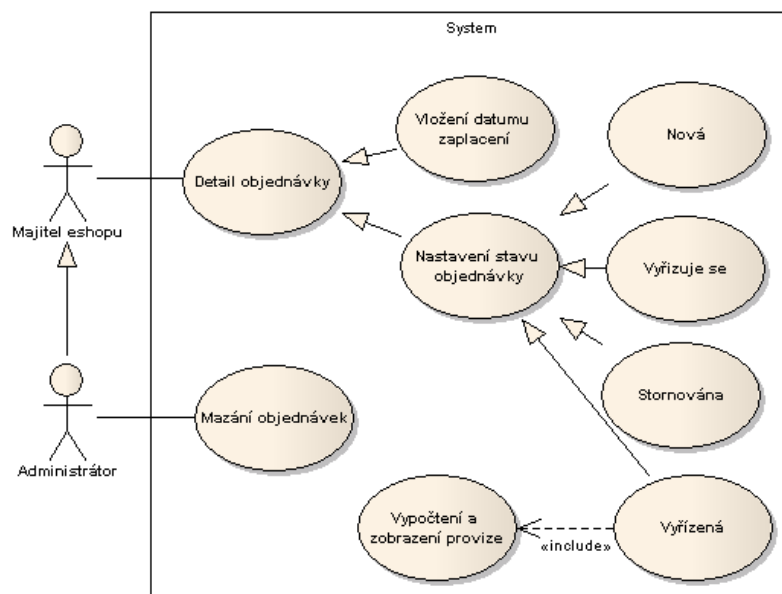
12. Export je povolen
13. Systém přenesení úlohu do fronty úloh na pozadí, aby se dal web dále ovládat
14. Systém zjistí, že se jedná o první úlohu a nastaví jí stav běžící
15. Systém provede export zboží do xml souboru
16. Systém smaže danou úlohu z fronty úloh
17. Systém odešle informační zprávu o úspěchu uživateli

Rozšíření:

- 1a – Uživatel chce upravit vybraný export
 - 1a1 – Uživatel vybere patřičný export z výpisu exportů, skok na bod 3
- 3a – Systém načte formulář
 - 3a1 – Systém načte uložené údaje do formuláře
 - 3a2 – Uživatel upraví potřebné hodnoty, skok na bod 5
- 7a – Údaje nejsou v pořádku
 - 7a1 – Systém nahlásí uživateli chybu o nesprávném formátu nebo nevyplnění potřebných údajů, skok na bod 3
- 12a – Export není povolen
 - 12a1 – Systém informuje uživatele o neoprávněné operaci, skok na bod 9
- 14a – Úloha není první ve frontě
 - 14a1 – Systém nastaví této úloze stav čekající, skok na 14

5.1.4 Správa objednávek

Diagram na obrázku 15 zobrazuje Use Case diagram části správa objednávek. Ke správě objednávek mohou přistupovat oba hlavní aktéři systému a to jak majitel eshopu, tak hlavně administrátor. Ten může oproti majitelovi mazat objednávky. Oba si mohou zobrazit detail objednávky, nastavit stav objednávky na vyřizuje se, stornována nebo vyřízená. Nové objednávky jsou logicky označeny stavem nová. Ze všech objednávek, které jsou vyřízené a jejichž patro má nastavenou nějakou hodnotu provize se získává provize z objednávky.



Obrázek 15: Use Case – Správa objednávky

5.2 Specifikace chování

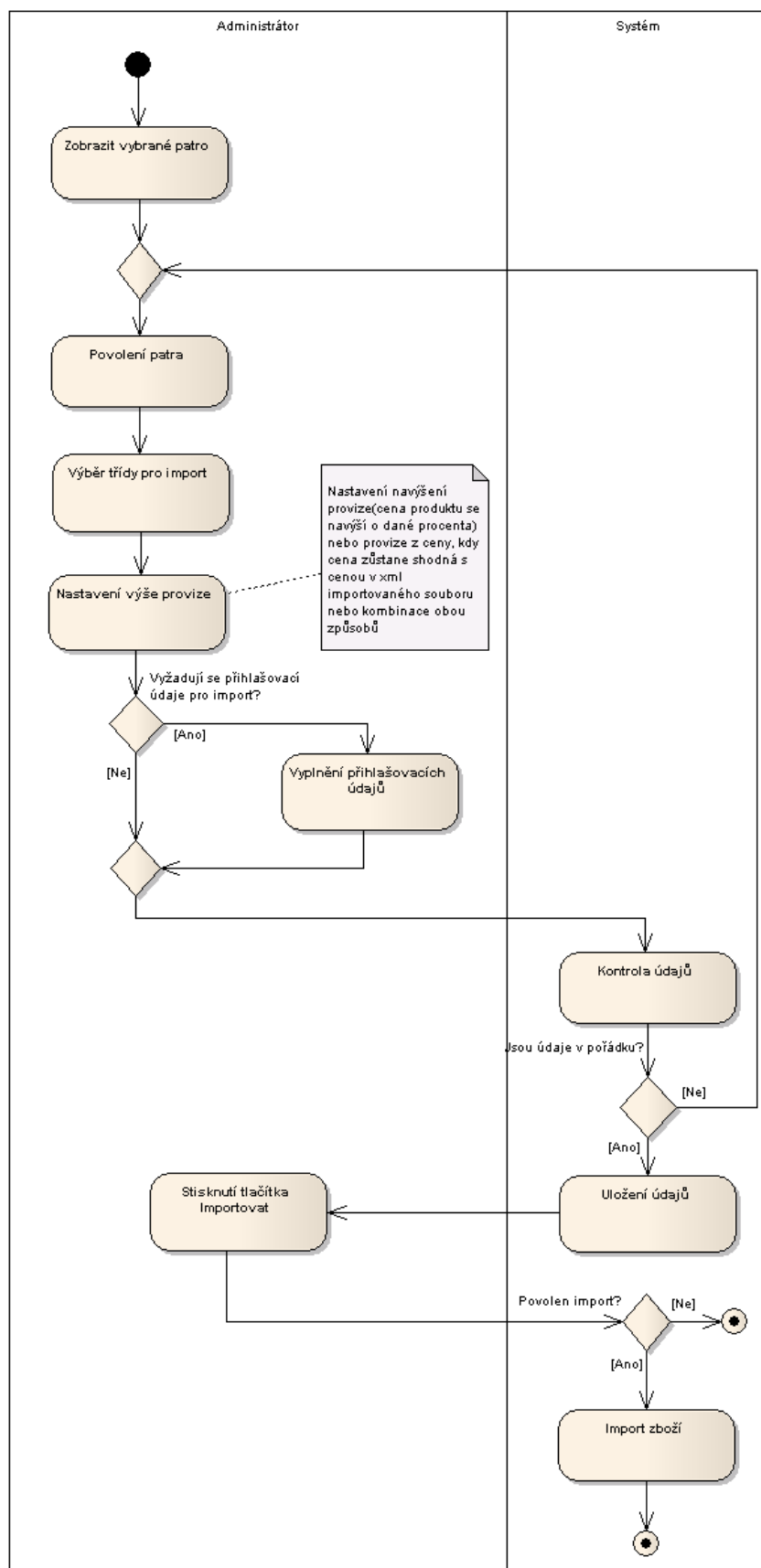
V této části si ukážeme specifikaci chování pomocí aktivitních diagramů, které se používají pro bližší popis toku činností daný vnitřním mechanismem jejich provádění [5].

Jelikož by bylo náročné popisovat a tvořit aktivitní diagramy pro celý systém, vybral jsem dva klíčové.

5.2.1 Import zboží

Na obrázku 16 je znázorněn proces importu zboží od distributora, vybraného patra.

Proces začíná vybráním patřičného patra, pro které chceme získat zboží. Pokud chceme provádět import zboží, musíme patro povolit. Dalším krokem je nastavení výše provize. Pokud jsou vyžadovány přihlašovací údaje pro stažení feedu, uživatel vyplní tyto údaje. Také se musí nastavit třída pro import, tj. třída, která je schopna xml soubor daného patra přeparsovat. Pokud jsme spokojeni, tak stiskneme tlačítko pro uložení údajů. Pokud jsou tyto údaje v pořádku, dojde k uložení údajů, v opačném případě se celý proces opakuje. Poslední aktivitou je stisknutí tlačítka „Importovat“ a tím se naimportuje zboží do vybraného patra.

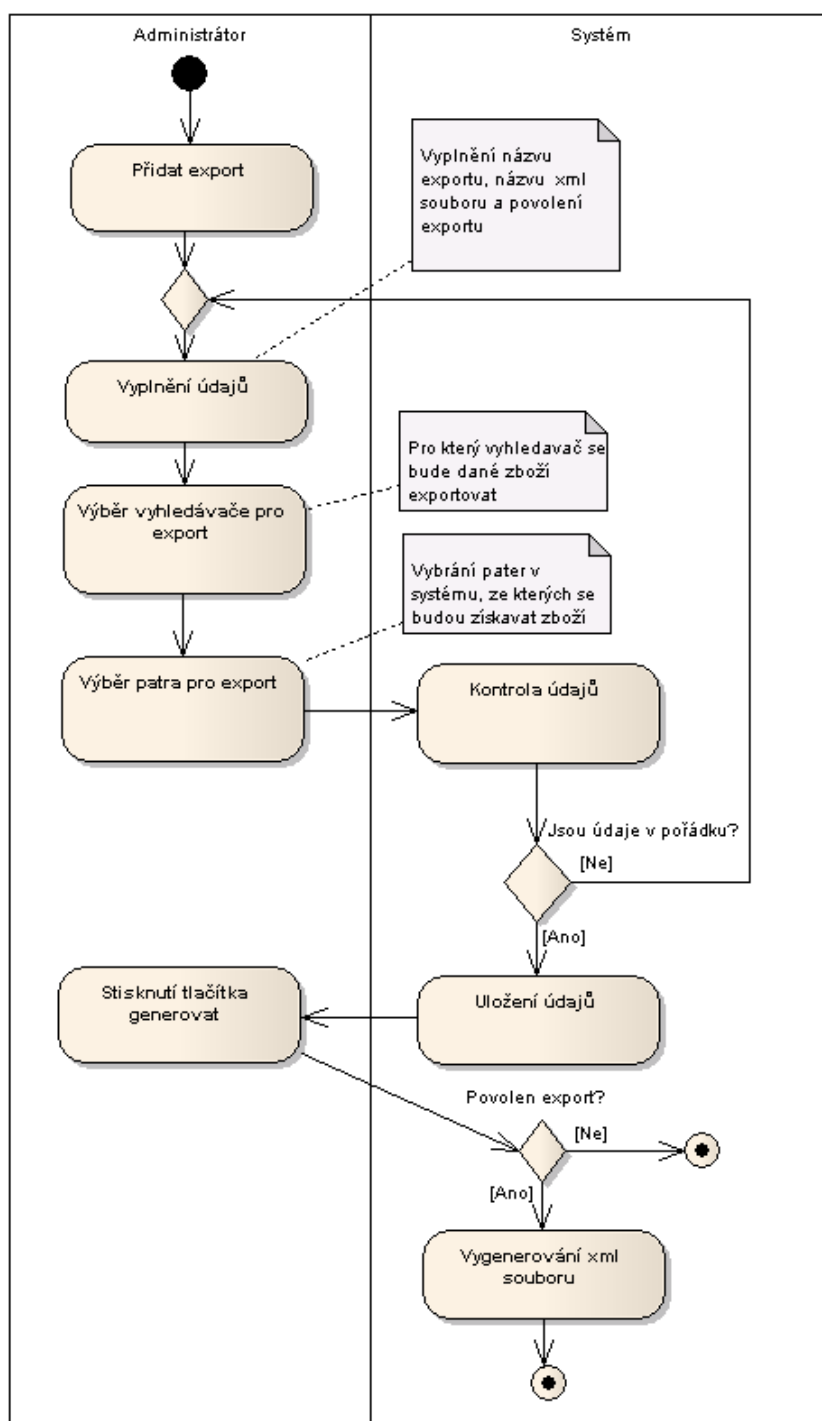


Obrázek 16: Aktivitní diagram – Import zboží

5.2.2 Export zboží do vyhledávačů

Na obrázku 17 je aktivitním diagramem znázorněn proces exportu zboží pro vyhledávače. Proces je zobrazen pomocí swimlane(plaveckých drah) pro názornost, kdo kterou část vykonává.

Proces začíná u uživatele, který chce přidat export. Uživatel vyplní údaje jako je název exportu, název xml souboru a povolí export. Vybere vyhledávač, pro který se bude zboží exportovat(zboží.cz, heureka, hyperzboží.cz apod.). Důležitou částí je výběr pater, jejichž zboží se bude exportovat. Systém provede kontrolu správnosti zadaných údajů. Pokud jsou údaje v pořádku, uloží tyto údaje. V opačném případě musí uživatel údaje opravit. Dalším krokem je stisknutí tlačítka „Exportovat“ uživatelem. Pokud je export aktivní, dojde k vygenerování xml souboru pro vybraný vyhledávač.



Obrázek 17: Aktivitní diagram – Export zboží do vyhledávačů

6 Analýza a návrh

Cílem toku činností zabývajícím se analýzou a návrhem softwarového produktu je ukázat jakým způsobem bude produkt realizován v implementační fázi. Model analýzy, zkoumá specifikované požadavky z pohledu objektů, které lze nalézt v problémové doméně. Model návrhu dále upřesňuje model analýzy ve světle skutečného implementačního prostředí. [5]

6.1 Datová analýza

Je součástí návrhu databázového systému a je to volba pro vhodné uložení dat, definici tabulek, jejich atributů a vztahů mezi těmito tabulkami. Hlavním cílem datového modelování je navrhnout datovou strukturu pro konkrétní aplikaci a databázový systém, který se bude používat pro uložení dat této aplikace.

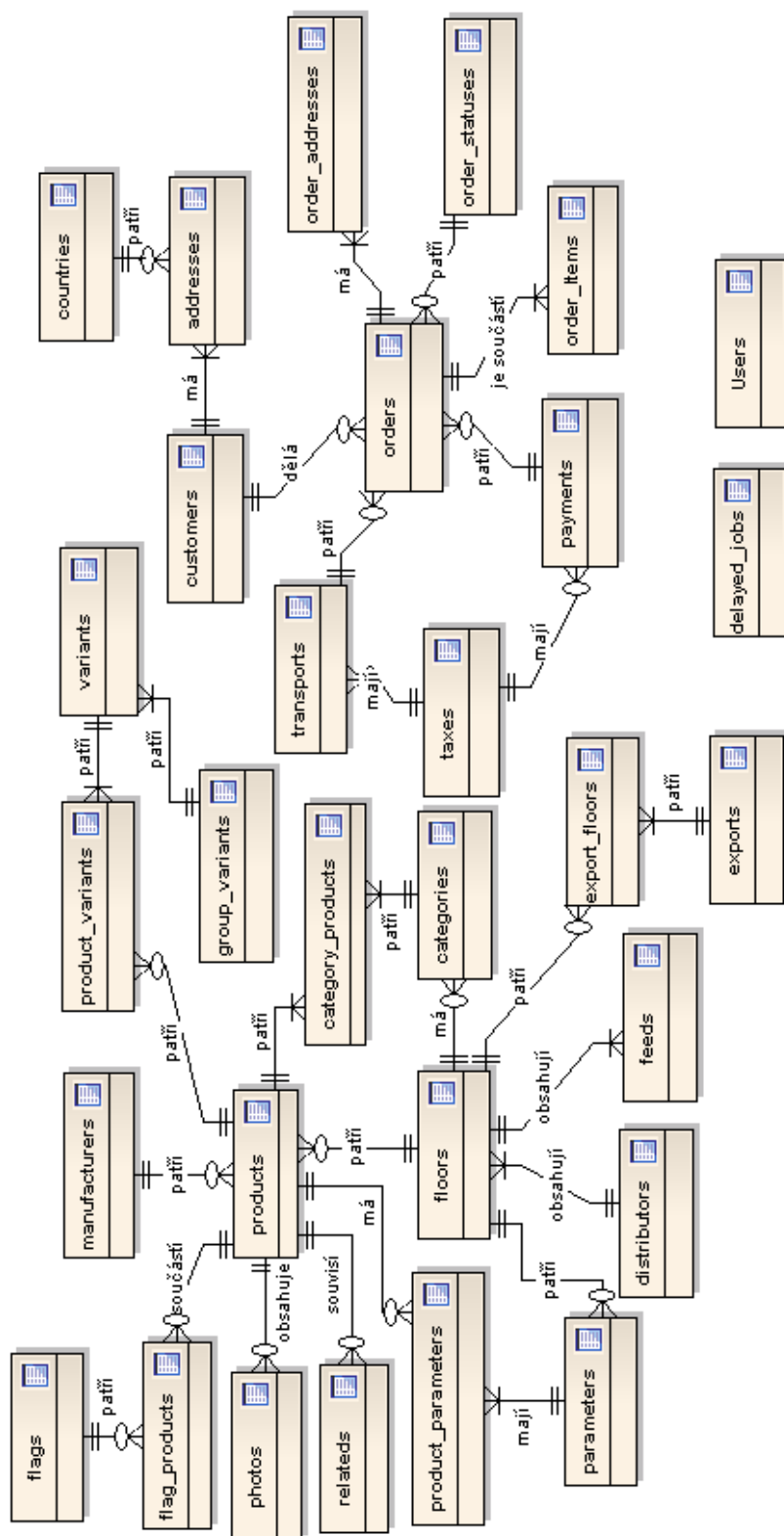
V této práci se zaměřím na konceptuální datový model, jehož záměrem je popis dat v databázi, nezávisle na fyzickém uložení těchto dat v této databázi.

Jednou ze součástí datové analýzy je ER diagram. Protože se snažím veškeré, v této práci prezentované, diagramy kreslit pomocí standardu UML, použil jsem pro kresbu ER diagramu konceptuální datový model, který vychází z jazyka UML. Výsledný datový model můžeme vidět na obrázku 18.

Nezbytnou částí datové analýzy je datový slovník. Ten se skládá z jednotlivých typů entit (názvy tabulek) a jejich atributů. U každého atributu se eviduje jakého je datového typu daný atribut, jestli se jedná o primární nebo sekundární klíč, zda může být atribut nulový a jestli bude označen jako index. Protože datový slovník obsahuje mnoho tabulek, najdete ho v příloze C.

6.2 Statická struktura

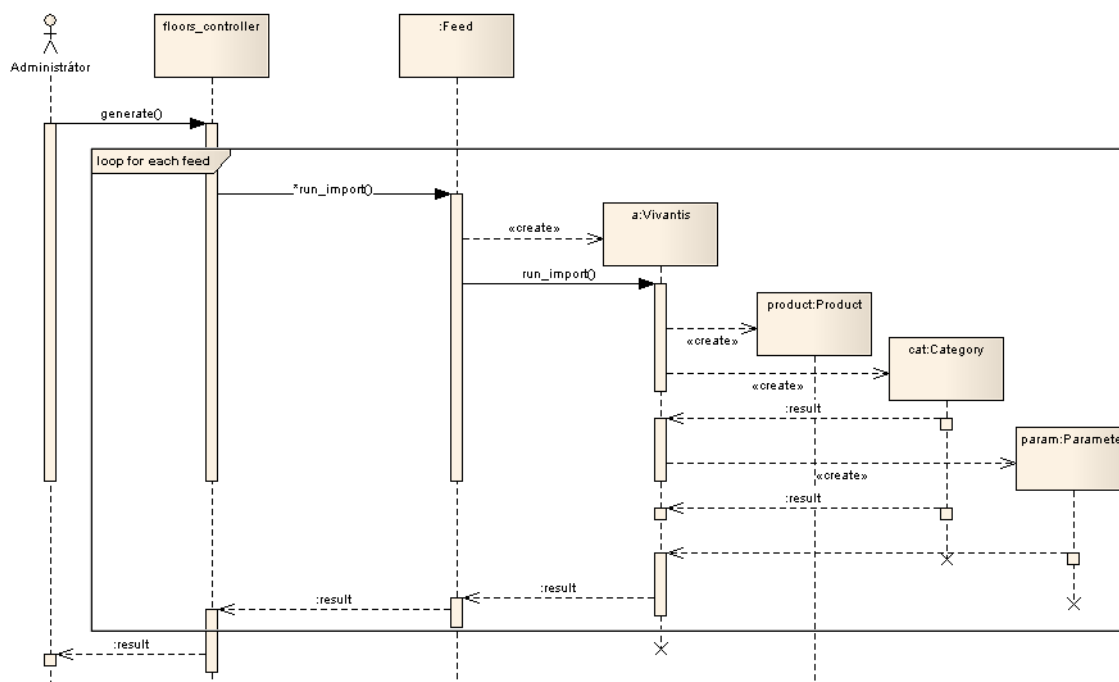
Statická struktura systému specifikuje jeho prvky, vnitřní strukturu těchto prvků a vztahy mezi nimi. Tuto strukturu popíšu třídním diagramem, který popisuje obecně platné třídy a typy, jejich vnitřní strukturu a vzájemné relace, nezávislé na čase. [6] Třídní diagram vidíme na obrázku 19.



Obrázek 18: Datový model



Obrázek 19: Třídní diagram



Obrázek 20: Sekvenční diagram – Import zboží

6.3 Dynamická struktura

V předchozí části jsme si ukázali jak pomocí třídního diagramu udělat statickou strukturu, v této části si ukážeme jak zachytit dynamické chování. Hlavní podstatou je modelování komunikace mezi objekty. Takovéto iterace poté vedou k požadovanému chování celého systému tj. k zajištění jeho funkcionality. Tyto iterace popíšeme pomocí sekvenčních diagramů. Jelikož není smyslem této práce popsat všechny iterace, ukážeme si jejich smysl jen na části import zboží.

6.3.1 Import zboží

Na obrázku 20 vidíme sekvenční diagram, znázorňující proces importu zboží od dodavatelů. Komunikace mezi objekty začíná posláním zprávy generate controlleru floors. Tato zpráva se vytvoří, pokud uživatel stiskne tlačítko „Importovat“. Controller ve smyčce prochází všechny feedy, patřící vybranému patru a postupně je spouští. Spuštění feedu znamená poslání zprávy run_import modelu Feed. Model Feed poté vytvoří objekt třídy, které zpracuje daný xml soubor z Feedu. V našem případě to bude třída Vivantis. Vytvořený objekt spustí metodu run_import, která začne parsovat xml soubor.

Parsování probíhá postupným vytvořením objektů Product, Category a Parameter, do kterých se uloží data z xml souboru.

Pokud patro obsahuje další xml soubor, tak se celý proces opakuje. Pokud je proces parsování konečný, zašle se informační zpráva o úspěchu uživateli.

V následující části si ukážeme jak je systém navržen a jakou používá strukturu.

6.4 Model-View-Controller

V současné době se hodně při vývoji softwaru preferuje vícevrstvá architektura systému. Mezi tu nejznámější, a v tomto systému hlavní architektu, patří třívrstvá architektura Model-View-Controller. Znamená to, že aplikace je rozdělena do vrstev, čímž odděluje funkční, zobrazovací a vykonávací logiku. Modifikace jedné části nemá vliv na části další. Struktura každé Ruby on Rails aplikace je postavena na této struktuře. Hlavní výhoda je v tom, že při vývoji systému na něm spolupracuje více lidí od programátorů po grafiky. Grafici a lidé, co dělají vzhled stránek samozřejmě nemusí ovládat programovací jazyk. Stačí jen jednoduché kaskádové styly a menší základy s HTML. Tím, že je architektura oddělená, mohou tito lidé pracovat na pohledech a další část lidí může tvořit logiku systému. Nyní si popíšeme jednotlivé části architektury Model-View-Controller. [1]

Model - je datovým základem celé aplikace, který zajišťuje zpracování dat. Představuje logiku aplikace a stará se o ukládání, načítání, aktualizace dat. Rovněž se stará o různé výpočty nad daty. Nemá ponětí o okolním prostředí a neví nic o svém běhu na webovém serveru, jen mu předáváme údaje a řekneme mu, co se má provést. Model zpracuje data a vrátí nám výsledek. Model se v Ruby on Rails také stará o validace nad daty (správnost vstupů). V Ruby on Rails je model třídou, která zprostředkovává kontakt s databází.

View - je zodpovědný za zobrazení výsledku akce. Je to jednoduše řečeno klasická webová stránka, kterou vidíme na obrazovce s definovanými ovládacími prvky jako jsou tlačítka, odkazy apod. K datům pohled přistupuje přes model, který je zpracuje. Například máme seznam produktů a ten je přístupný přes model, ale je to pouze pohled, který tento list zobrazí uživateli ve formě webové stránky.

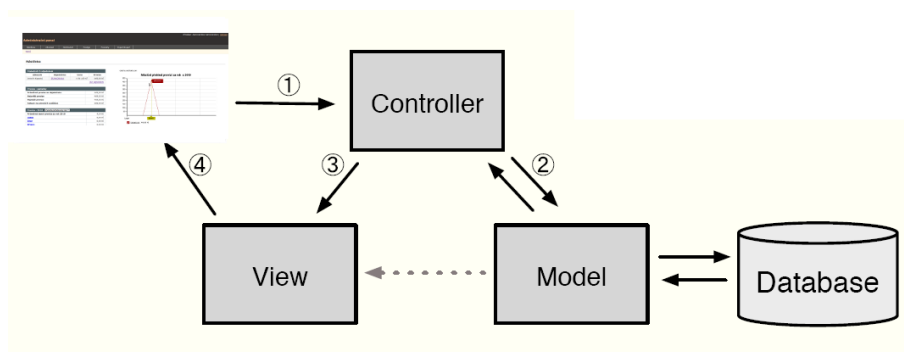
Controller - organizuje aplikaci. Zpracovává požadavky od klienta a určuje, jak se s nimi bude zacházet a distribuuje výstupy. Určuje, který model se má pro zobrazení použít. Dokáže trasovat požadavky mezi jednotlivými akcemi.

Na obrázku 21 vidíme strukturu MVC.

1. Prohlížeč posílá požadavek
2. Controller interaguje s modelem, který manipuluje s daty v databázi
3. Controller vyvolá view
4. View vyrenderuje obraz

6.5 ActiveRecord

Jak jsme si řekli v kapitole 3 ActiveRecord je srdcem každé Rails aplikace. Je to objektově-relační vrstva, dodávaná s Rails. Tato vrstva mapuje tabulky na třídy, řádky tabulky



Obrázek 21: Architektura Model-View-Controller

na objekty a sloupce mapuje na atributy. Liší se od mnoho dalších objektově-relačních vrstev, způsobem, jakým je konfigurován. Spoléhá na konvenci a minimalizuje počet konfigurací, které musí programátor vykonat. ActiveRecord podporuje validace modelových dat. Jakmile nastane chyba v nějakém formuláři, Rails pěkně zobrazí chybu, přičemž nám stačí pouze jeden řádek v modelu. Je to pevný základ MVC architektury Ruby on Rails aplikace. [1]

6.6 Bezpečnost

Mezi důležitou částí této práce bude patřit zabezpečení před neoprávněným vstupem. Uživatel, který nemá přístup do systému nebo nemá přístup k nějaké části se tam nesmí dostat i když adresu napíše přímo do URL. Proto každá taková akce bude mít varovné upozornění a dojde k přesměrování na úvodní stránky.

Navíc bude myšleno, aby při editaci nebo zobrazení patřičné položky, kde je na konci URL identifikační číslo dané položky, nemohl uživatel toto číslo přepsat a dostat se do autorizované sekce. Jakmile toto číslo přepíše na neexistující ID, které se v systému nenachází např. 545, bude tento nepovolený režim patřičně ohlášen a dojde k přesměrování.

7 Implementace

Cílem implementace je doplnit navrženou architekturu (kostru) aplikace o programový kód a vytvořit tak kompletní systém. Implementační model specifikuje jak jsou jednotlivé elementy (objekty a třídy) vytvořené v etapě návrhu implementovány ve smyslu softwarových komponent, kterými jsou zdrojové kódy, spustitelné kódy, data a podobně. [5]

V následující části popíšu různé technické záležitosti, záludnosti a problémy, se kterými jsem se setkal při implementování aplikace.

7.1 Struktura kódu pro import a export dat

Na obrázku 22 je zobrazena struktura souborů, které jsou použity pro import a export dat. Každý soubor je třídou, která je umístěna do modulu. Modul je ohraničená část kódu, podobná definici třídy. Tyto moduly jsou umístěny v adresáři Libs.

Vidíme, že máme modul „merchant“, který má v sobě další moduly „imports“ a „exports“. V těchto modulech jsou jednotlivé třídy, pro zpracování buď importu zboží nebo exportu zboží do vyhledávačů, podle toho, v jakém modulu se daná třída nachází.

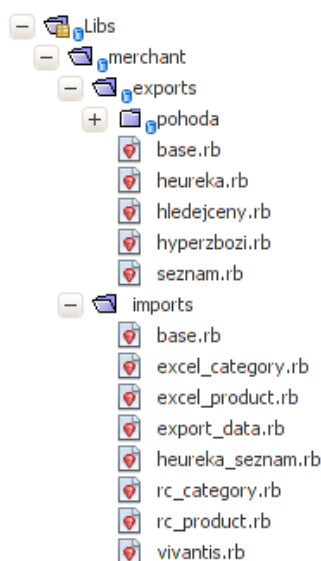
Každý z obou modulů má v sobě základní třídu „Base“, která je rodičovskou třídou ostatních tříd v modulu. Třída „Base“ má metody, společné všem jejím potomkům (ostatním třídám v modulu). Důležité metody třídy „Base“ jak modulu imports, tak exports popíšu v následující částech.

7.2 Import zboží

Import zboží, jak již samotný název napovídá, slouží k importu zboží od dodavatelů do systému. Import se provádí přes xml soubor, který se nachází na serveru dodavatele. Každý xml soubor všech dodavatelů obsahuje název produktu, ceny a obrázek produktu. Většina obsahuje kategorie, dostupnost, popisky a dodatečně může xml soubor obsahovat výrobce, související produkty, parametry produktů, různé značky jestli je produkt novinka, doporučený, v akci, potom varianty zboží apod.

Jak jsem se zmínil v předchozí sekci, veškeré soubory, zpracovávající import, jsou umístěny v modulu „imports“. Základní třída „Base“ obsahuje následující důležité metody, které jsou sdíleny potomkům:

- **run_import** – metoda, která spouští postupně ostatní metody
- **construct_url** – zkonstruuje url adresu. Pokud např. dodavatel vyžaduje pro přístup k xml souboru přihlašovací údaje, tak url adresa, ze které stahuji xml soubor může vypadat následovně: `http://dodavatelsky-web.cz?zakaznik=jmeno&heslo=heslo`. Tyto přihlašovací údaje získám z nastavení patra.
- **download_xml** – stáhne tělo xml souboru (jeho strukturu) a uloží si ho do proměnné
- **presetup_revised** – abych mohl rozpoznávat, které produkty při importu již v xml souboru nejsou a měly by být smazány z databáze, používám u všech tabulek, které



Obrázek 22: Rozmístění souborů

souvisí se zbožím atribut *is_revised*, který je typu boolean. Metoda *presetup_revised* nastaví před každým importem atribut u všech tabulek, které souvisí se zbožím na hodnotu 0.

- **parse_xml** – metoda, která kompletně přeparsuje xml soubor a uloží jeho hodnoty do patřičných tabulek v databázi. Každá položka má pole *is_revised* např. (*product_is_revised*, *manufacturer_is_revised* apod.), kde si ukládám všechny id položek (produkty, kategorie, výrobce, související produkty, varianty apod.), které jsem přeparsoval a jsou tak v xml souboru.
- **postsetup_revised** – v metodě *parse_xml*, jak jsem se zmínil, mám pole ideček, všech položek, které jsem navštívil a jsou v xml souboru. Tato metoda projde všechny tabulky a nastaví všem řádkům, které mají id v tom poli hodnotu *is_revised* na hodnotu 1. Tím mám označené všechny produkty, které jsou v xml souboru a jsou tak aktuální.
- **destroy_obsolete** – metoda, která smaže všechny položky z tabulek, které mají nastavenou hodnotu *is_revised* na hodnotu 0. Tyto položky tudíž nebyly obsaženy v xml souboru a již nejsou aktuální a musí se smazat.
- **generate_time** – nastaví datum posledního importu na aktuální datum.

Potomci třídy „Base“ jsou třídy, které zpracovávají již konkrétní xml soubor. všechny třídy překryjí metodu *construct_url* protože každý dodavatel může mít jiný formát url adresy, pokud zadávají přihlašovací údaje, a metodu *parse_xml*, která nám přeparsuje xml soubor pro konkrétního dodavatele. Proto je také přetížená, protože každý dodavatel

může mít jinou strukturu xml. Pokud má dodavatel strukturu xml, která ještě není v systému, vytvoří se na to nová třída, která umí daného distributora přeparsovat.

V systému je zatím 8 tříd, které umí dosud zpracovat 17 pater (obchodů různého zaměření). Třída Vivantis zpracovává všechny obchody společnosti Vivantis, kterých je 6. Ostatní třídy jsou pro další obchody s odlišnou strukturou. Nejsložitější ze všech tříd jsou třídy ExcelCategory a ExcelProduct. Pro ně je formát dodáván ve formě MS Excel XML souboru, takže musím odlišovat atributy na prvním řádku a hodnoty na dalších řádcích, které přísluší těmto atributům. Je to řešené pomocí pole, kde si ukládám indexy jednotlivých sloupců a poté průchodem řádku kontroluji, jestli mám v poli daný sloupec a jak se jmenuje.

Samotný import dat probíhá tak, že si vezmu záznam tabulky feeds, která obsahuje url adresy xml souboru, pozice a důležitou položku import.type. Tento důležitý atribut odkazuje na třídu, která se má použít při importu dat.

Některé třídy umí parsovat jen kategorie. Tyto třídy jsou z důvodu, že někteří dodavatelé mají xml soubor pro kategorie a xml soubor pro produkty a ostatní věci. Samozřejmě je velice důležité, aby se kategorie importovaly před zpracováním produktů, protože produkty na ně odkazují. To řeším v nastavení tabulky feeds, kde mám atribut position. XML soubory se tak zpracovávají od nejnižší hodnoty atributu position (url xml s kategorií bude mít position 1 a xml s produkty position 2).

Důležitou věcí pro import produktů je nastavení provize. Pokud má tak patro nastavenou hodnotu „Navýšení provize“ na hodnotu větší než nula, musí se cena importovaného produktu před uložením do databáze zvýšit o nastavené procenta. Ostatní věci a výpočty provize nás při importu nezajímají.

Aktualizace záznamů se provádí tak, že se vyhledá jednoznačně identifikovatelný kód produktu. Pokud není v databázi, tak se přidá, jinak se záznam aktualizuje kompletním updatem údajů. Je to rychlejší, než kontrolovat každý atribut, jestli je jiný než v xml souboru.

Při zpracování importu jsem narazil na jeden velký problém. Při programování a testování jsem zkoušel parsování jen na 5ti výrobcích, abych nemusel dlouho čekat, než se proces dokončí. Jakmile jsem se ujistil, že třída pracuje jak má, zkusil jsem nahrát kompletně celé patro (např. 5tisíc produktů). Parsování těch 5tisíc produktů trvalo kolem 50 minut a to jsem neměl ani povolené stahování obrázku. Bylo to způsobeno obrovským množstvím přístupů do databáze. Problém jsem naštěstí vyřešil pomocí transakcí, které Railsy zvládají. Tím se mi zmenší počet přístupů do databáze na jeden a zároveň to vyřešilo problém, když nastane chyba při importu, pokud transakce neprojde celá, nic se do databáze neuloží. Nevzniká tak problém s neúplnými daty. Výsledek použití transakcí vypadá tak, že se mi čas z 50min změnil na pěkných 7min.

Kód na obrázku 23 ukazuje ukázkou importu kategorií pomocí transakce.

7.3 Export zboží

Export zboží slouží k exportu našich produktů do vyhledávačů. Systém spolupracuje s vyhledávači zboží.cz, heureka, hyperzbozi.cz, hledej ceny.cz.

```

#import kategorii
ActiveRecord::Base.transaction do
  CATEGORIES_FIELDS.each_pair do |k, v|
    self.raw_xml.find(v).each do |item|
      unless item.content == ""
        self.revised_categories << Category.find_or_create_by_name_and_floor_id_and_xml_id(item.content, @floor.id, item["id"]).id
      end
    end
  end
end
end

```

Obrázek 23: Import kategorií pomocí transakce

Veškeré soubory, zpracovávající export, jsou umístěny v modulu „exports“. Základní třída „Base“ obsahuje následující důležité metody, které jsou sdíleny potomkům:

- **run_export** – metoda, která spouští postupně ostatní metody
- **generate** – vygeneruje xml soubor, obsahující všechny povinné a volitelné údaje pro vyhledávače
- **postsetup_export** – nastaví atribut `is_altered` na 0. Pokud je atribut `is_altered` na hodnotě 1, došlo k updatu záznamů v administračním rozhraní. Nastavením na hodnotu 1 se mi url adresa v administračním rozhraní stává neviditelnou. To upozorní uživatele že je potřeba stisknout tlačítko „Exportovat“, aby byla adresa přístupná

U exportu je důležité vybrat třídu pro export, která bude export, podobně jako třída pro import, zpracovávat. Výběr třídy je v administračním rozhraní, kde vybereme, jaký vyhledávač použít. Třídy se jmenují stejně jako vyhledávač.

Export zboží, stejně jako import zboží se v administračním rozhraní odehrává na pozadí. Jakmile uživatel stiskne tlačítko, proces se zařadí do fronty úloh, pokud není první, jinak se spustí. To mi umožňuje s aplikací dále pracovat i když běží nějaká úloha. Bylo odvěcí vytvořit nějakého menšího „správce úloh“, který by mi zobrazil přehled úloh ve frontě a běžící úlohy. Jinak by uživatel nepoznal, že se něco odehrává. Úlohy, které jsou ve stavu čekající se mohou z fronty kdykoliv smazat a tím se zruší naplánovaný proces. Běžící úloha se musí dokončit.

Vygenerovaný xml soubor ukládám na server aplikace, takže bude dostupný na url http://lukaskamp.gansa.skvely.org/export/xml_soubor.xml.

7.4 Zpracování provizí

Poslední a neméně důležitá část implementační fáze je zpracování provizí. Provize je, jak již bylo zmíněno v kapitole o affiliate marketingu, odměna za zprostředkování prodeje.

Provize nastavuji u každého patra, podle toho, jaký typ provize dodavatel nabízí. Ve specifikaci požadavků jsem vysvětlil, že jsou tři druhy provizí:

- **navýšení provize** kdy se mi zvýší cena prodávaného zboží o dané procento provize a můj výdělek za zboží získám odečtením staré ceny produktu od nové


```

#vrati celkovou hodnotu provize z objednávky
def price_bonus
  price = 0
  if status == 4
    order_items.each do |item|
      price = item.quantity * item.product_price

      if item.increase_bonus > 0 and item.bonus_price > 0
        #kombinace
        price_before = (100 * price) / (item.increase_bonus + 100)
        price_increase_bonus = price - price_before
        price_bonus_price = (price_before * item.bonus_price)/100
        price += price_bonus_price + price_increase_bonus
      elsif item.increase_bonus > 0
        #navyseni ceny produktu pri importu
        price_before = (100 * price) / (item.increase_bonus + 100)
        price += price - price_before
      elsif item.bonus_price > 0
        #provize z ceny
        price += (item.bonus_price * price)/100
      end
    end
  end
  price
end

```

Obrázek 24: Provize z objednávky

- **provize z ceny** kdy cena produktu se nezmění, výdělek spočítám vynásobením ceny produktu hodnotou provize(jako desetinné číslo), viz. kapitola specifikace požadavků 5.
- **kombinace obou způsobů**

Při importu si tak stačí dát pozor, jestli je nastavena provize „navýšení provize“ nebo kombinace, aby se zvýšila cena importovaného produktu.

Veškeré výpočty ohledně provizí jsou umístěny v metodách modelu „Order“, tedy modelu, který se stará o objednávky. Při zpracování provizí je důležité rozlišovat, jestli je distributor plátce DPH nebo ne. Provize počítám jako ceny produktů bez DPH dané objednávkou. Pokud je distributor plátce DPH, musí se k vypočtené provizi připočíst ještě ona daň za zprostředkování obchodu. Pokud plátce DPH není, již se nic nepřičítá. Metodu pro výpočet provize z objednávky vidíme na obrázku 24. Provize se počítá jen u objednávek, které mají nastaven stav na vyřízená (status 4). Poté procházím všechny produkty z objednávky a kontroluji, jakou mají nastavenou provizi. Pokud je hodnota > 0 u navýšení provize(increase.bonus), počítá se jen tato provize, pokud je hodnota provize z ceny > 0 počítá se zase jen provize z ceny(bonus.price). Pokud jsou vyplněné obě hodnoty, počítá se kombinace. Vidíme, že u produktů, které mají nastaven typ provize „navýšení provize“, musím počítat s hodnotou před navýšením, abych ji mohl poté od nové hodnoty ceny produktu odečíst.

Měsíční provizi spočítám jako součet hodnot provizí z jednotlivých objednávek za vybraný měsíc. Do výpočtu jsem zapojil také výpočet průměrné denní provize a průměrné roční provize.

Průměrná denní provize za celý rok se spočítá jako součet provizí ze všech měsíců a podělím počtem dní v roce. Tady musím rozlišovat, jestli jedná o aktuální rok nebo již

```

#prumerna denni provize za rok
def self.average_bonus_daily_year(year)
  days_in_year = 0
  sum_bonus_month = 0
  #pokud se jedna o aktualni rok, zjistí pocet dni do tohoto datumu
  if year == Time.now.year
    1.upto(Time.now.month-1) do |i|
      sum_bonus_month += bonus_month(i, year)
      start_date = DateTime.parse("#{i}/01/#{year}")
      end_date = start_date+1.month-1.seconds
      days_in_year +=end_date.day
    end
    sum_bonus_month += bonus_month(Time.now.month, year)
    days_in_year += Time.now.day
  else
    12.times do |i|
      sum_bonus_month += bonus_month(i+1,year)
      start_date = DateTime.parse("#{i+1}/01/#{year}")
      end_date = start_date+1.month-1.seconds
      days_in_year +=end_date.day
    end
  end
  sum_bonus_month / days_in_year
end

```

Obrázek 25: Průměrná denní provize za rok

uplynulý rok, protože pokud se jedná o aktuální rok a je např. leden, nemohu započítávat počet dní v roce 365 ale počet dní, které uplynulo od začátku roku do aktuálního data. Pokud je to již některý předešlý rok, tak stačí podělit počtem dní v tom roce. Metodu pro výpočet vidíme na obrázku 25.

Podobně se spočítá průměrná denní provize za vybraný měsíc. Součet objednávek za onen měsíc, podělím počtem dní v tom měsíci. Opět musím rozlišit, jestli už měsíc uplynul nebo je aktuální. Pokud je to běžící měsíc, tak nepočítám počet dní v tom měsíci, ale počet dní, které uběhly od začátku měsíce po dnešní datum.

Veškerý přehled provizí zobrazuji na hlavní stránce po přihlášení do administračního rozhraní.

7.5 Webové standardy

Jsou v současném vývoji internetových stránek důležitou věcí. Web postavený na webových standardech by měl být jednoduchý, bezchybný, založený na CSS, přístupný, použitelný a přívětivý k vyhledávačům. Dodržováním těchto standardů si zefektivníme svou práci a současně budeme dodávat stránky, které jsou přístupné širšímu okruhu lidí. Výhodou dodržování standardů je, že se stránky se budou rychleji načítat a budou se správně zobrazovat i v příštích verzích prohlížečů, které tyto standardy dodržují. Organizace, která se stará o tyto standardy se nazývá W3C [12].

Tato aplikace se snaží tyto standardy dodržovat. Obsahuje layout, který nevyužívá pro jeho tvorbu tabulky, je postavený na kaskádových stylech CSS. Stránky jsem se snažil udělat plně validní, pro bezproblémový chod ve všech moderních internetových prohlížečích. Stránky byly v pořádku v prohlížečích Google Chrome, Firefox a Opera. Problém byl samozřejmě v InternetExploreru, který né úplně standardy dodržuje. Problém

byl naštěstí jen v chybném zobrazení vysouvacího menu. Stačil naštěstí menší hack a problém byl vyřešen.

7.6 Přístupný web

Přístupný web se dá charakterizovat jako web, který neklade žádné překážky na uživatele stránek. Stejně jako můžeme mít přístupné budovy pro různě hendikepované uživatele, např. vozíčkáře, můžeme mít přístupný web pro uživatele s různými zrakovými problémy, s omezenou pohyblivostí apod. který neomezuje tyto uživatele web používat. Přístupný web je vlastně typem bezbarierovosti.

Jelikož všichni uživatelé na Internetu nejsou stejní, někteří neslyší, nemluví, mají omezenou pohyblivost nebo špatně vidí nebo nevidí vůbec, snažíme se dělat stránky i pro tento typ uživatelů.

Přístupnost se nemusí týkat jen uživatelů s fyzickým hendikepem. Můžeme mezi tyto uživatele zařadit uživatele se slabším HW vybavením, pomalým přístupem na Internet apod.

Tato aplikace se nezabývá internetovým obchodem jako takovým, neřeší frontend část pro zákazníky, proto přístupnost v části administrace není až tak důležitá jako v prostoru pro zákazníky. Přeci jen budou mít k této části přístup pouze uživatelé, kteří budou mít správu shopu na starosti.

Přesto i v této části se dali použít některé prvky přístupnosti, jako je např. pěkný a čitelný formát url adresy. Všichni to známe, ty dlouhé url adresy, které si normální uživatel nemůže zapamatovat. Pro přístupnost jsou takové adresy nevhodné. Proto mají veškeré adresy líbivý a lehce čitelný formát.

Na stránkách rovněž používám zobrazení, ve které části stránek se uživatel zrovna nachází, pro přehlednost, a může se rovněž přepnout na předchozí úroveň.

Podstatnou věcí, kterou jsem se snažil při vývoji dodržet bylo příjemné a lehce pochopitelné a ovladatelné uživatelské rozhraní. V aplikaci tak nejsou různé pasti pro uživatele, aby nevěděl co má vlastně dělat a kde se nachází určitá část.

7.7 Ajax

Je pojem o kterém jde v současné době hodně slyšet. Zkratka AJAX znamená Asynchronous JavaScript and XML a je to označení pro technologie vývoje interaktivních webových aplikací, které mění svůj obsah bez nutnosti znovu načítání celé stránky [9]. Máme-li na stránce nějaký element, ve kterém máme třeba seznam videí a pod tímto elementem máme odkaz na další stránku s podobnými videi, po kliknutí se nám aktualizuje pouze tento box a nemusí se načítat kompletně celá webová stránka jako menu, hlavní obsah, když nebyla změněna. To urychluje načítání stránek. Navíc můžeme přidávat různé efekty při přepínání, jako zvýraznění, postupné mizení apod.

V aplikaci jsem to s Ajaxem moc nepřeháněl. Použil jsem ho prakticky jen na hlavní stránce, kde po rozbalení měsíce překresluje obsah elementu pro zobrazení objednávky v daném měsíci. Také se používá na stránce se seznamem spuštěných procesů, kde se mi

periodicky překresluje tabulka procesů. Pěkně tak vidím, kdy úlohy ve frontě skončí – zmizí, aniž bych musel zkoušet jednou za čas refresh stránky.

8 Závěr

V této práci jsem se seznámil s principy fungování Affiliate marketingu a na základě toho jsem vypracoval analýzu a návrh celého jádra affiliate internetového obchodu. Poté přišla na řadu fáze implementační, kde jsem celé jádro implementoval v technologii Ruby on Rails.

Práce na této aplikaci byla velice zajímavá a přinesla mi mnoho nových zkušeností a poznatků, hlavně s vývojem v technologii Ruby on Rails. Tato technologie je mi velice sympatická a rád bych v ní pokračoval i po vystudování, protože si myslím že má budoucnost. Jak již bylo zmíněno v úvodu, tato práce byla vytvořena pro firmu Skvělý.CZ s.r.o, která by chtěla aplikaci využívat ke svým obchodním účelům. Spolupráce s touto firmou byla bezproblémová a byla velká snaha zodpovědět každou otázku, případně pomoci s nějakým problémem. Velkou zásluhu na tom mají pan Ing. Jiří Kubica, který je jednatel firmy a který mě vedl teoreticky a potom pan Ing. Robert Cigán, hlavní vývojář Ruby on Rails, který mi pomáhal s praktickou částí a ukazoval různé možnosti využití plně objektového jazyka Ruby a frameworku Ruby on Rails. Díky němu jsem posunul své znalosti toho frameworku o hodně výše. Oba tito pánové mě donutili vytvořit aplikaci, se které jsem měl popravdě na začátku velké obavy, jestli jsem schopen to zvládnout. Donutili mě studovat a řešit problémy, takže jsem strávil mnoho hodin hledáním různých řešení a postupů na internetu. Tím jsem získal větší přehled o Ruby on Rails jako takovém.

V práci jsem se snažil využít moderních internetových standardů, pro tvorbu současných webových aplikací. Hlavně jsem se snažil dodržet validitu celé aplikace pro bezproblémový chod ve všech moderních internetových prohlížečích, využití layoutu stránky, který nepoužívá tabulky a zapojení AJAX technologie. Požadavkem bylo rovněž vytvořit intuitivní a příjemné ovládání celého systému.

Věřím, že těch několik stovek až tisíců hodin, které jsem nad vývojem aplikace strávil, nebyly nadarmo.

Lukáš Kamp

9 Reference

- [1] Dave Thomas a David Heinemeier Hansson *Agile Web Development with Rails: Second Edition, Third Edition*, The Pragmatic Programmers LLC., 2009
- [2] Dave Thomas *Programming Ruby: The Pragmatic Programmer's Guide, Second Edition*, The Pragmatic Programmers LLC., 2004
- [3] Steven Holzner *Začínáme programovat v Ruby on Rails*, Computer Press, a.s., 2007
- [4] Karmi is on Rails - weblog o Ruby On Rails a web designu, dostupný na url adrese <http://blog.karmi.cz>
[online] [cit. 2009-06-25]
- [5] Prof. Ing. Ivo Vondrák, CSc. *Úvod do softwarového inženýrství*, 2004
- [6] Prof. Ing. Ivo Vondrák, CSc. *Metody specifikace softwarových systémů*, 2005
- [7] Wikipedie, otevřená encyklopedie – MySQL <http://cs.wikipedia.org/wiki/MySQL>
[online] [cit. 2010-04-25]
- [8] Wikipedie, otevřená encyklopedie – Javascript <http://cs.wikipedia.org/wiki/Javascript>
[online] [cit. 2010-04-25]
- [9] Wikipedie, otevřená encyklopedie – AJAX <http://cs.wikipedia.org/wiki/AJAX>
[online] [cit. 2010-04-25]
- [10] Wikipedie, otevřená encyklopedie – XHTML
http://cs.wikipedia.org/wiki/Extensible_HyperText_Markup_Language
[online] [cit. 2010-04-25]
- [11] Wikipedie, otevřená encyklopedie – CSS
http://cs.wikipedia.org/wiki/Cascading_Style_Sheets
[online] [cit. 2010-04-25]
- [12] World Wide Web Consortium, dostupný na url adrese <http://www.w3.org>
[online] [cit. 2010-04-25]

A Uživatelská příručka

V této části si popíšeme základní práci s aplikací, která je dostupná na adrese:
<http://lukaskamp.gansa.skvely.org/admin>

A.1 Uživatelské role systému

- **Administrátor** - má na starosti kompletní správu celého systému, má tak veškerá přístupová práva ke všem sekcím v systému. Sekce se týkají správy dopravy, platby, daní, zobrazení produktů, úprava dodavatelů a pater, import zboží, přidání, úprava a export zboží, zobrazení provizí systému, správa objednávek, správa uživatelů. V neposlední řadě může zobrazovat úlohy ve frontě naplánovaných úloh a mazat úlohy, které neběží, ale jsou ve frontě připraveny k běhu.

Přihlašovací údaje: *login: admin, heslo: heslo*

- **Majitel eshopu** – může zobrazovat detaily produktů systému, zobrazovat hodnoty svých provizí na hlavní stránce systému a nastavovat stav a datum zaplacení u objednávek. Do dalších sekcí systému nemá přístupová práva.

Přihlašovací údaje: *login: demo, heslo: heslo*

A.2 Administrační rozhraní systému

V této části se seznámíme s ovládání celé aplikace. Pro přihlášení do systému použijeme výše zmíněné přístupové údaje, podle uživatelské role, pod kterou se chceme přihlásit.

A.2.1 Nástěnka

Okamžitě po přihlášení se zobrazí hlavní stránka se seznamem provizí na levé straně a grafy, zobrazující měsíční přehled provizí za vybraný rok, koláčový graf se stavy objednávek a graf zobrazující počet objednávek za daný měsíc vybraného roku. Stránku jsem nazval „Nástěnka“ a je vidět na obrázku 26.

Na levé straně máme přehled posledních 5ti objednávek, pokud chceme vidět ostatní objednávky, použijeme buď tlačítko „více objednávek“ nebo menu „Prodeje -> Objednávky“.

Následují statistiky ohledně provizí. Jedná se o přehled průměrných provizí za objednávku, nejvyšší a nejnižší hodnota provize a poté celková hodnota provize za všechny měsíce a roky.

Hlavní část přehledu spočívá v zobrazení měsíců, u kterých okamžitě vidíme hodnotu provize za daný měsíc. Po rozkliknutí vybraného měsíce dojde k zobrazení objednávek, kterých se provize týká. Rovněž nechybí statistický údaj, průměrná denní provize. Nepostradatelnou součástí je změna roku, pro který chceme získat data. Změnou roku se

Posledních 5 objednávek			
Zákazník	Objednávka	Cena	Provize
Antonín Kopecký	28/04/2010/7	13 267,00 Kč	0,00 Kč
Antonín Kopecký	28/04/2010/6	13 267,00 Kč	692,36 Kč
Petr Novák	28/03/2010/5	36 276,00 Kč	773,18 Kč
Petr Novák	28/02/2010/4	27 771,00 Kč	0,00 Kč
Antonín Kopecký	28/01/2010/1	4 811,00 Kč	469,00 Kč
více objednávek			

Provize - statistiky	
Průměrná provize za objednávku	645,00 Kč
Nejvyšší provize	773,18 Kč
Nejnižší provize	469,00 Kč
Celkem na provizích vyděláno	1 935,00 Kč

Provize - 2010 Vyberte požadovaný rok ▼			
Průměrná denní provize za rok 2010	16,39 Kč		
Leden	469,00 Kč		
Průměrná denní provize	15,13 Kč		
Zákazník	Objednávka	Cena	Provize
Antonín Kopecký	28/01/2010/1	4 811,00 Kč	469,00 Kč

Únor	0,00 Kč
Březen	773,00 Kč
Duben	692,00 Kč
Květen	0,00 Kč
Červen	0,00 Kč
Červenec	0,00 Kč
Srpen	0,00 Kč
Září	0,00 Kč
Říjen	0,00 Kč
Listopad	0,00 Kč

chart by amCharts.com

Měsíční přehled provizí za rok a 2010

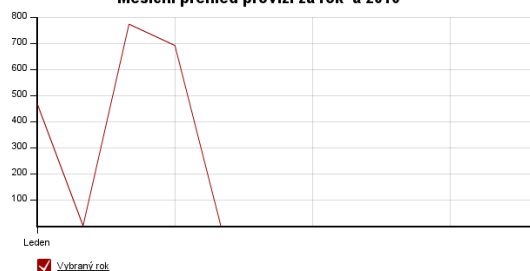
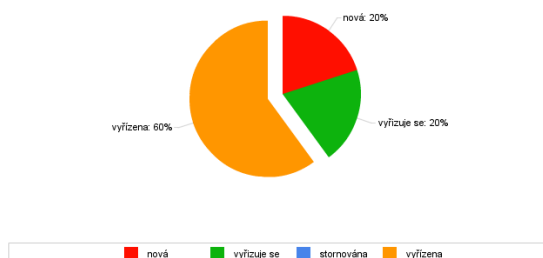


chart by amCharts.com

Přehled stavů objednávek



Obrázek 26: Nástěnka s přehledem provizí

změní tabulku s měsíčním přehledem provizí a všechny grafy na pravé straně se překreslí podle vybraného roku.

A.2.2 Správa uživatelů

Správa uživatelů slouží k přidávání, editování a mazání uživatelů, kteří se mohou přihlásit do systému administračního rozhraní. Tuto činnost může provádět jen administrátor systému. Administrátor nemůže smazat sám sebe. Formulář pro vložení nového uživatele vidíme na obrázku 27.

A.2.3 Distributoři

Pokud v úpravě informací odškrtneme možnost „Povolit distributora“ potom distributor zmizí i z menu importy a exporty. Tím nebude povolen import dat od tohoto distributora. stejně tak jako nebude možnost exportovat produkty z patra tohoto distributora.

A.2.4 Objednávky

Přehled všech objednávek najdeme v menu „Prodeje – > Objednávky“. K objednávkám má přístup jak majitel eshopu, tak samozřejmě administrátor. Pouze administrátor může objednávky mazat. Na obrázku 28 vidíme výpis všech objednávek. U objednávek, které jsou uzavřené a které mají produkty z patra, mající nastavenou provizi, se zobrazuje

Nový uživatel

Osobní informace	
Jméno: *	<input type="text"/>
Příjmení: *	<input type="text"/>
Uživatelské jméno: *	<input type="text"/> <small>Minimum jsou 3 znaky</small>
E-mail:	<input type="text"/>
Nastavení uživatelského hesla	
Heslo: *	<input type="text"/> <small>Minimum je 5 znaků</small>
Potvrzení hesla: *	<input type="text"/>

Obrázek 27: Formulář pro vložení nového uživatele

Přihlášen: Administrátor Administrátor | [Odhlásit](#)

Administrační panel

Nástěnka	Uživatelé	Distributoři	Prodeje	Produkty	Import/Export
----------	-----------	--------------	---------	----------	---------------

[Domů](#) > Objednávky

Objednávka byla úspěšně uložena

Objednávky

[Exportovat objednávky do Pohody](#)

Vybrat: Vše Nic	Zaškrtnuté: <input checked="" type="checkbox"/>						
Číslo objednávky	Datum objednání	Datum zaplacení	Zákazník	Cena s DPH	Stav objednávky	Provize s DPH	Akce
<input type="checkbox"/> 28/04/2010/6	28. 04. 2010 06:21:48	21. 04. 2010 00:00:00	Antonín Kopecký	13 267,00 Kč	vyřízena	692,36 Kč	
<input type="checkbox"/> 28/04/2010/5	28. 04. 2010 06:21:30	29. 04. 2010 00:00:00	Petr Novák	36 276,00 Kč	vyřízena	773,18 Kč	
<input type="checkbox"/> 28/04/2010/4	28. 04. 2010 06:14:55		Petr Novák	27 771,00 Kč	vyřizuje se	0,00 Kč	
<input type="checkbox"/> 28/04/2010/1	27. 04. 2010 22:54:03	30. 04. 2010 00:00:00	Antonín Kopecký	4 811,00 Kč	vyřízena	469,00 Kč	

Obrázek 28: Výpis všech objednávek

hodnota provize za danou objednávku. Objednávky, které se změnili po dobu 14 dní se budou po stisknutí patřičného tlačítka, exportovat do ekonomického informačního systému Pohoda.

K detailu objednávky se dostaneme kliknutím na řádek vybrané objednávky nebo kliknutím na ikonu pera napravo. Detail objednávky může vypadat jako na obrázku 29. Objednavce se dá nastavit stav objednávky a datum zaplacení objednávky. Pokud má objednávka nastaven stav vyřízená a obsahuje produkty z patra, které mají nastavenou nějakou výši provize, potom se zobrazí obsah provize u takovéto objednávky. Uživatel může rovněž vložit nějakou poznámku pro zákazníka.

Objednávka č. 28/04/2010/6

Uložit objednávku

Objednávka 28/04/2010/6		Informace o účtu zákazníka	
Datum objednání	28. 04. 2010 06:21:48	Jméno zákazníka	Antonín Kopecký
Stav objednávky	nová	Email	antonin.kopecky@gmail.com
Doprava	Česká pošta	Registrován	27. 04. 2010 22:54:03
Platba	Hotově		
Zaplaceno			
Provize	0,00 Kč		

Fakturační adresa		Dodací adresa	
Společnost s.r.o. Komenského 12 Brno, 12345		Lukáš Tuček Masarykova 45 Praha, 54873	
IČ: CZ125415	DIČ: DIC12541	IČ:	DIČ:
Telefon: 774 804 512		Telefon: 747 878 789	

Poznámka od zákazníka		Poznámka pro zákazníka	

Položky objednávky			
Název produktu	Cena za položku s DPH	Množství	Cena s DPH
Chrono 16299/2	4 690,00 Kč	1 ks	4 690,00 Kč
HK5420-313	840,00 Kč	1 ks	840,00 Kč
Coenzystar Q10 (Cardiol Q 10)	850,00 Kč	1 ks	850,00 Kč
Anatomax (Anatomix)	638,00 Kč	1 ks	638,00 Kč
Achyrocline satureioides - čaj	213,00 Kč	1 ks	213,00 Kč
Thymunex	2 934,00 Kč	1 ks	2 934,00 Kč
BETULIC - Fin Betutabs	275,00 Kč	1 ks	275,00 Kč
ACEROLA XYLISWEET - FIN Xylacertabs	545,00 Kč	1 ks	545,00 Kč
Extra Cell Guard (Cascading Revanol)	815,00 Kč	1 ks	815,00 Kč
Brain Star (Ingenious)	1 346,00 Kč	1 ks	1 346,00 Kč

Součty objednávky	
Poštovné a balné s DPH	121,00 Kč
Celkem bez DPH	11 735,42 Kč
Celkem s DPH	13 267,00 Kč

Obrázek 29: Detail objednávky

MAMBO-CZ s.r.o.					
Káva a čaj	Ne	27. 04. 2010 17:36:52	27. 04. 2010 17:36:52		
Radim Janda					
Alu kola	Ne	27. 04. 2010 17:36:52	27. 04. 2010 17:36:52		
Tojs s.r.o.					
Hračky	Ne	27. 04. 2010 17:36:52	27. 04. 2010 17:36:52		
ÚTM s.r.o.					
RC modely	Ne	27. 04. 2010 17:36:52	27. 04. 2010 17:36:52		
VIVANTIS a.s.					
Hodinky	Ano	27. 04. 2010 17:36:51	27. 04. 2010 17:37:21		Importovat
Kosmetika	Ne	27. 04. 2010 17:36:51	27. 04. 2010 17:36:51		
Lékárna	Ano	27. 04. 2010 17:36:51	28. 04. 2010 06:17:03		Importovat
Módní doplňky	Ne	27. 04. 2010 17:36:51	27. 04. 2010 17:36:51		
Parfémy	Ne	27. 04. 2010 17:36:51	27. 04. 2010 17:36:51		
Šperky	Ne	27. 04. 2010 17:36:51	27. 04. 2010 17:36:51		

Obrázek 30: Správa importů

A.2.5 Daně, dopravy a platby

Slouží k nastavení způsobů dopravy objednávky, platby objednávky a daní, které se vybírají jak u dopravy, tak i platby.

A.2.6 Produkty

Seznam všech produktů, které jsme importovali do systému. Produkty jsou seříděné podle abecedy v pořadí distributor, patro, název produktu. Pokud chceme vidět detail produktu, klikneme na vybraný řádek produktu.

A.2.7 Importy

Stránka s importy je nejdůležitější věc celého systému. Slouží ke správě pater, které se budou importovat. Na obrázku 30 vidíme přehled Distributorů a pater, které distributoři nabízejí. Také vidíme, jestli je patro povolené, kdy bylo naposledy generováno. Napravo spatříme tlačítko „Importovat“, které zahájí import patra.

Pokud klikneme na distributora, dostaneme se na stránku s detailem distributora. Pokud vybereme patro, zobrazí se formulář s nastavením informací o patru. Detail patra obsahuje nastavení povolení, jestli se bude dané patro importovat - „Povolení patra“. Lze také nastavit krátký a dlouhý popis patra.

Velice důležitou věc v nastavení patra je provize. Jak již bylo několikrát zmíněno, jsou 2 typy provize, které nastavujeme. Podrobněji viz. kapitola 5 Na obrázku 31 vidíme možnost nastavení provize. Pokud nastavíme provizi a dáme import, produkty se naimportují do našeho eshopu s nastavenou hodnotou provize.

Veškerý import patra, jak již bylo zmíněno, se odehrává na pozadí. Je to z důvodu ovladatelnosti webu i v době importu. Celý proces můžeme sledovat ve stejném menu, tentokrát procesy.

Typ a výše provize	
Navyšení provize (%):	<input type="text" value="5"/> <small>Hodnota, o kolik procent se navýší cena importovaného výrobku</small>
Provize z ceny (%):	<input type="text" value="5"/> <small>Udává hodnotu provize, při které se cena importovaného produktu nezmění</small>
Provize celkem:	10 %

Přihlašovací údaje pro import	
Přihlašovací jméno:	<input type="text"/>
Přihlašovací heslo:	<input type="password"/>
Potvrzení hesla:	<input type="password"/>

Obrázek 31: Nastavení provize v detailu patra

Informace o exportu	
Název exportu: *	<input type="text" value="Export pro zboží.cz"/>
Povolení exportu:	<input checked="" type="checkbox"/>

XML soubor	
Jméno souboru: *	<input type="text" value="seznam.xml"/> <small>Název souboru zadávájte včetně přípony .xml</small>

Vyhledávač pro export	
Vyhledávač: *	<input type="text" value="seznam"/>

Patra pro export	
<input checked="" type="checkbox"/> Hodinky	
<input type="checkbox"/> Léčárna	

Obrázek 32: Přidání exportu

A.2.8 Exporty

Pokud budeme chtít exportovat produkty do vyhledávačů, použijeme k tomu menu „Exporty“. Exporty můžeme vkládat, editovat a mazat. Formulář s údaji o exportu vidíme na obrázku 32. Nějak si export pojmenujeme, povolíme exportování, pojmenujeme si vytvořený xml soubor, včetně koncovky *.xml. Musíme vybrat vyhledávač, pro který bude export použit. Nesmíme také zapomenout na výběr patra, ze kterého se budou exportovat produkty.

Po uložení se dostaneme na stránku s přehledem všech exportů. Po stisknutí tlačítka „Exportovat“ se export spustí, rovněž na pozadí a poté je vytvořen xml soubor. Ihned po vygenerování se mezi výpisy objeví odkaz na tento xml soubor, který poté můžeme zaregistrovat do vyhledávačů zboží. Pokud provádíme úpravu exportu, vidíme, že odkaz na xml soubor zmizí. Je to z důvodu, aby to donutilo uživatele ihned po úpravě, kdy můžeme změnit exportované patra, kliknout na tlačítko „Exportovat“, pro zachování aktuálnosti údajů.

Přihlášen: Administrátor Administrátor | [Odhlásit](#)

Administrační panel

Nástěnka

Uživatelé

Distributoři

Prodeje

Produkty

Import/Export

[Domů](#) > Spuštěné procesy

Spuštěné procesy

Tabulka se aktualizuje automaticky

ID procesu	Stav	Typ procesu	Spuštění procesu	Uzamčení procesu	Pokus spuštění	Akce
3	běžící	Import - Patro: Lékárna, Dodavatel: VIVANTIS a.s.	28. 04. 2010 06:19:35	28. 04. 2010 06:19:39	0	
4	čekající	Export - Export pro seznam	28. 04. 2010 06:19:40		0	✖

Obrázek 33: Procesy

A.2.9 Procesy

Pokud budeme data importovat nebo exportovat, zvolil jsem provádět tyto akce na pozadí, z důvodu ovladatelnosti webu během těchto činností. Aby uživatel viděl, že se něco děje, vytvořil jsem stránku s přehledem spuštěných a naplánovaných procesů. Vzhled stránky můžeme vidět na obrázku 33.

Jednotlivé spuštěné úlohy se řadí do fronty. Mám dva druhy stavu procesů a to běžící proces a čekající proces. Čekající proces můžu smazat s úlohy, stisknutím křížku. Běžící proces již nejde zrušit. Celá tabulka procesů se aktualizuje automaticky každou sekundu, takže není potřeba refreshovat stránku. Pokud proces dokončí svou úlohu, z tabulky jednoduše zmizí.

B Obsah přiloženého CD

Součástí této diplomové práce je přiložené CD obsahující:

- text této diplomové práce v elektronické podobě (ve formátu PDF a TEX)
- zdrojové kódy aplikace
- programátorskou dokumentaci, vygenerovanou pomocí rdoc (obdoba javadoc pro Ruby on Rails)

C Datový slovník

Users – tabulka uživatelů, přihlašujících se do backend rozhraní

Atribut	Datový typ	Klíč	Null	Index	Popis
id	int(11)	PK	N	Y	id uživatele
login	varchar(255)	N	N	N	login uživatele
encrypted_password	varchar(255)	N	N	N	kryptované heslo
salt	varchar(255)	N	N	N	řetězec, kterým je heslo šifrováno
firstname	varchar(255)	N	N	N	jméno uživatele
lastname	varchar(255)	N	N	N	příjmení uživatele
email	varchar(255)	N	Y	N	email uživatele
is_admin	boolean	N	N	N	je administrátor
created_at	datetime	N	N	N	datum vytvoření
updated_at	datetime	N	N	N	datum úpravy

Tabulka 1: Users

Distributors – tabulka distributorů

Atribut	Datový typ	Klíč	Null	Index	Popis
id	int(11)	PK	N	Y	id distributora
name	varchar(255)	N	N	Y	název distributora
city	varchar(255)	N	N	N	město
street	varchar(255)	N	N	N	ulice
postcode	varchar(255)	N	N	N	psč
telephone	varchar(255)	N	Y	N	telefon
email	varchar(255)	N	Y	N	email
web	varchar(255)	N	Y	N	webové stránky
company_id	varchar(255)	N	Y	N	IČ
vat_number	varchar(255)	N	Y	N	DIČ
description	text	N	Y	N	doplňkový text
created_at	datetime	N	N	N	datum vytvoření
is_enabled	boolean	N	N	N	povolení distributora

Tabulka 2: Distributors

Feeds – tabulka feedů, obsahující jednotlivé xml feedy

Atribut	Datový typ	Klíč	Null	Index	Popis
id	int(11)	PK	N	Y	id feedu
floor_id	int(11)	FK	N	Y	id patra
feed_url	varchar(255)	N	N	N	url feedu
import_type	varchar(255)	N	N	N	typ třídy pro import
position	int(11)	N	N	N	pořadí při zpracování feedu(pro více feedu v patře)
created_at	datetime	N	N	N	datum vytvoření
updated_at	datetime	N	N	N	datum úpravy

Tabulka 3: Feeds

Floors – tabulka pater

Atribut	Datový typ	Klíč	Null	Index	Popis
id	int(11)	PK	N	Y	id patra
distributor_id	int(11)	FK	N	Y	id distributora
name	varchar(255)	N	N	Y	název patra
permalink	varchar(255)	N	N	Y	permalink
short_description	varchar(255)	N	Y	N	krátké informace
description	text	N	Y	N	informace k patru
is_enabled	boolean	N	N	N	povolení patra
increase_bonus	int(11)	N	N	N	navýšení provize
bonus_price	int(11)	N	N	N	provize z ceny
login	varchar(255)	N	Y	N	přihlašovací jméno pro xml
password	varchar(255)	N	Y	N	heslo pro xml
last_generate	datetime	N	Y	N	čas posledního generování
start_time	datetime	N	Y	N	čas zahájení importu
frequency	varchar(255)	N	N	N	frekvence opakování
created_at	datetime	N	N	N	datum vytvoření
updated_at	datetime	N	N	N	datum úpravy

Tabulka 4: Floors

Categories – tabulka kategorií

Atribut	Datový typ	Klíč	Null	Index	Popis
id	int(11)	PK	N	Y	id kategorie
floor_id	int(11)	FK	N	Y	id patra
parent_id	int(11)	FK	N	Y	id nadkategorie
name	varchar(255)	N	N	Y	název kategorie
permalink	varchar(255)	N	N	Y	permalink
xml_id	int(11)	N	N	Y	id v xml souboru
is_enabled	boolean	N	N	N	povolení kategorie
is_revised	boolean	N	N	N	před každým importem se kategorie (daného feedu) nastaví na 0. Aktualizované, nezměněné nebo přidané kategorie jsou nastaveny na 1. Kategorie s 0 se smažou

Tabulka 5: Categories

Category_Products – vazební tabulka mezi kategoriemi a produkty

Atribut	Datový typ	Klíč	Null	Index	Popis
id	int(11)	PK	N	Y	id
category_id	int(11)	FK	N	Y	id kategorie
product_id	int(11)	FK	N	Y	id produktu
is_revised	boolean	N	N	N	1 – přidaný, nezměněný, aktualizovaný 0 – můžu smazat
floor_id	int(11)	N	N	Y	id patra

Tabulka 6: Category_Products

Export_Floors – vazební tabulka mezi patry a exportem

Atribut	Datový typ	Klíč	Null	Index	Popis
id	int(11)	PK	N	Y	id
floor_id	int(11)	FK	N	Y	id patra
export_id	int(11)	FK	N	Y	id exportu

Tabulka 7: Export_Floors

Exports – tabulka exportů

Atribut	Datový typ	Klíč	Null	Index	Popis
id	int(11)	PK	N	Y	id exportu
export_type	varchar(255)	N	N	N	třída pro export
name	varchar(255)	N	N	N	název exportu
filename	varchar(255)	N	N	N	název xml souboru
path	varchar(255)	N	Y	N	url adresa ke složce exportu
last_generate	datetime	N	Y	N	datum posledního generování
is_enabled	boolean	N	N	N	povolení exportu
is_altered	boolean	N	N	N	Změněn export? Pro nezobrazení xml souboru při úpravě hodnot
start_time	datetime	N	N	N	čas zahájení exportu
frequency	varchar(255)	N	N	N	frekvence opakování
created_at	datetime	N	N	N	datum vytvoření
updated_at	datetime	N	N	N	datum úpravy

Tabulka 8: Exports

Products – tabulka produktů

Atribut	Datový typ	Klíč	Null	Index	Popis
id	int(11)	PK	N	Y	id produktu
manufacturer_id	int(11)	FK	N	Y	id výrobce
floor_id	int(11)	FK	N	Y	id patra
name	varchar(255)	N	N	Y	název produktu
description	text	N	Y	N	popisek produktu
long_description	text	N	Y	N	dlouhý popisek produktu
code	varchar(255)	N	N	Y	kód produktu
price	decimal(10,5)	N	N	N	cena produktu
price_vat	decimal(10,0)	N	N	N	cena s DPH
vat	decimal(10,0)	N	N	N	daň
availability	varchar(255)	N	Y	N	dostupnost ve dnech
delivery_of	varchar(255)	N	Y	N	doba dodání od
delivery_to	varchar(255)	N	Y	N	doba dodání do
permalink	varchar(255)	N	N	Y	permalink
is_revised	boolean	N	N	N	Před každým importem se produkty (daného feedu) nastaví na 0. Aktualizované, nezměněné nebo přidané produkty jsou nastaveny na 1. Produkty s 0 se smažou
is_enabled	boolean	N	N	N	povolení zobrazení produktu
created_at	datetime	N	N	N	datum vytvoření
updated_at	datetime	N	N	N	datum úpravy

Tabulka 9: Products

Manufacturers – tabulka výrobců

Atribut	Datový typ	Klíč	Null	Index	Popis
id	int(11)	PK	N	Y	id výrobce
name	varchar(255)	N	N	Y	název výrobce
floor_id	int(11)	N	N	Y	id patra
is_revised	varchar(255)	N	N	N	1 – přidaný, nezměněný, aktualizovaný 0 – můžu smazat
xml_id	int(11)	N	N	Y	id v xml souboru

Tabulka 10: Manufacturers

Relateds – tabulka pro související produkty

Atribut	Datový typ	Klíč	Null	Index	Popis
id	int(11)	PK	N	Y	id relateds
product_id	int(11)	FK	N	Y	id produktu
related_id	int(11)	FK	N	Y	id souvisejícího produktu
is_revised	boolean	N	N	N	1 – přidáný, nezměněný, aktualizovaný 0 – můžu smazat
floor_id	int(11)	N	N	Y	id patra

Tabulka 11: Relateds

Photos – tabulka obrázků pro produkty

Atribut	Datový typ	Klíč	Null	Index	Popis
id	int(11)	PK	N	Y	id obrázku
product_id	int(11)	FK	N	Y	id produktu
photo_file_name	varchar(255)	N	N	N	jméno souboru
photo_content_type	varchar(255)	N	N	N	typ obrázku
photo_file_size	int(11)	N	N	N	velikost obrázku
photo_updated_at	datetime	N	N	N	datum uploadu obrázku
photo_remote_url	varchar(255)	N	N	N	url adresa importovaného obrázku
photo_change	varchar(255)	N	Y	N	datum změny obrázku na serveru
is_revised	boolean	N	N	N	1 – přidáný, nezměněný, aktualizovaný 0 – můžu smazat
floor_id	int(11)	N	N	Y	id patra
is_main	boolean	N	N	N	1 – hlavní obrázek, 0 – další obrázky

Tabulka 12: Photos

Parameters – tabulka s pojmenovanými parametry k produktům

Atribut	Datový typ	Klíč	Null	Index	Popis
id	int(11)	PK	N	Y	id parametru
name	varchar(255)	N	N	N	název parametru

Tabulka 13: Parameters

Products_Parameters – vazební tabulka s hodnotami parametrů k produktům

Atribut	Datový typ	Klíč	Null	Index	Popis
id	int(11)	PK	N	Y	id hodnoty parametru
product_id	int(11)	FK	N	Y	id produktu
parameter_id	int(11)	FK	N	Y	id parametru
value	varchar(255)	N	N	N	hodnota parametru
floor_id	int(11)	N	N	Y	id patra
is_revised	boolean	N	N	N	1 – přidáný, nezměněný, aktualizovaný 0 – můžu smazat

Tabulka 14: Products_Parameters

Customers – tabulka registrovaných zákazníků

Atribut	Datový typ	Klíč	Null	Index	Popis
id	int(11)	PK	N	Y	id zákazníka
firstname	varchar(255)	N	N	N	jméno zákazníka
lastname	varchar(255)	N	N	N	příjmení zákazníka
email	varchar(255)	N	N	N	email zákazníka
login	varchar(255)	N	N	N	přihlašovací jméno
crypted_password	varchar(255)	N	N	N	kryptované heslo
salt	varchar(255)	N	N	N	řetězec, kterým je heslo šifrováno
created_at	datetime	N	N	N	datum registrace
updated_at	datetime	N	N	N	datum úpravy

Tabulka 15: Customers

Addresses – tabulka adres pro registrované zákazníky

Atribut	Datový typ	Klíč	Null	Index	Popis
id	int(11)	PK	N	Y	id adresy
customer_id	int(11)	FK	N	Y	id zákazníka
country_id	int(11)	FK	N	Y	id státu
firstname	varchar(255)	N	N	N	jméno
lastname	varchar(255)	N	N	N	příjmení
city	varchar(255)	N	N	N	město
street	varchar(255)	N	N	N	ulice
postcode	varchar(255)	N	N	N	psč
telephone	varchar(255)	N	N	N	telefon
company	varchar(255)	N	Y	N	společnost
company_id	varchar(255)	N	Y	N	IČ
vat_number	varchar(255)	N	N	N	DIČ
is_billing_address	boolean	N	N	N	je adresa fakturační?
is_delivery_address	boolean	N	N	N	je adresa dodací?

Tabulka 16: Addresses

Countries – tabulka států

Atribut	Datový typ	Klíč	Null	Index	Popis
id	int(11)	PK	N	Y	id státu
name	varchar(255)	N	N	N	název státu

Tabulka 17: Countries

Orders – tabulka objednávek

Atribut	Datový typ	Klíč	Null	Index	Popis
id	int(11)	PK	N	Y	id objednávky
customer_id	int(11)	FK	N	Y	id zákazníka
payment_id	int(11)	FK	N	Y	id platby
transport_id	int(11)	FK	N	Y	id dopravy
status	int(11)	N	N	N	stav objednávky 1-nová, 2-vyřizuje se, 3-stornována, 4-vyřízena
number	varchar(255)	N	N	N	identifikační číslo objednávky
comment_customer	text	N	Y	N	komentář zákazníka
comment_seller	text	N	Y	N	komentář prodáváče
price_transport	decimal(10,0)	N	N	N	cena dopravy s DPH
price_payment	decimal(10,0)	N	N	N	cena platby s DPH
vat_transport	decimal(10,0)	N	N	N	daň dopravy
vat_payment	decimal(10,0)	N	N	N	daň platby
paid	datetime	N	Y	N	datum zaplacení objednávky
processed	datetime	N	Y	N	datum zpracování
settled	datetime	N	Y	N	datum vyřízení
canceled	datetime	N	Y	N	datum stornování
created_at	datetime	N	N	N	datum vytvoření
updated_at	datetime	N	N	N	datum úpravy

Tabulka 18: Orders

Taxes – tabulka daní

Atribut	Datový typ	Klíč	Null	Index	Popis
id	int(11)	PK	N	Y	id daně
value	varchar(255)	N	N	N	hodnota daně v procentech

Tabulka 19: Taxes

Transports – tabulka způsobů dopravy

Atribut	Datový typ	Klíč	Null	Index	Popis
id	int(11)	PK	N	Y	id dopravy
tax_id	int(11)	FK	N	Y	id daně
name	varchar(255)	N	N	N	název dopravy
price_vat	decimal(10,0)	N	N	N	cena dopravy s DPH
description	text	N	Y	N	popisek dopravy

Tabulka 20: Transports

Payments – tabulka způsobů platby

Atribut	Datový typ	Klíč	Null	Index	Popis
id	int(11)	PK	N	Y	id platby
tax_id	int(11)	FK	N	Y	id daně
name	varchar(255)	N	N	N	název platby
price_vat	decimal(10,0)	N	N	N	cena platby s DPH
description	text	N	Y	N	popisek platby

Tabulka 21: Payments

Order Addresses – tabulka adres k objednávkce

Atribut	Datový typ	Klíč	Null	Index	Popis
id	int(11)	PK	N	Y	id adresy
order_id	int(11)	FK	N	Y	id objednávky
firstname	varchar(255)	N	N	N	jméno
lastname	varchar(255)	N	N	N	příjmení
city	varchar(255)	N	N	N	město
street	varchar(255)	N	N	N	ulice
postcode	varchar(255)	N	N	N	psč
country	varchar(255)	N	N	N	stát
company	varchar(255)	N	Y	N	společnost
company_id	varchar(255)	N	Y	N	IČ
vat_number	varchar(255)	N	Y	N	DIČ
telephone	varchar(255)	N	N	N	telefon
is_billing_address	boolean	N	N	N	je adresa fakturační?
is_delivery_address	boolean	N	N	N	je adresa dodací?

Tabulka 22: Order Addresses

Order_Items – tabulka pro položku objednávky

Atribut	Datový typ	Klíč	Null	Index	Popis
id	int(11)	PK	N	Y	id položky
order_id	int(11)	FK	N	Y	id objednávky
quantity	int(11)	N	N	N	množství daného produktu
product_name	varchar(255)	N	N	N	název produktu
product_price	decimal(10,5)	N	N	N	cena 1ks produktu
product_price_vat	decimal(10,2)	N	N	N	cena 1ks s DPH
product_vat	decimal(10,0)	N	Y	N	dph
increase_bonus	int(11)	N	Y	N	navýšení provize
bonus_price	int(11)	N	Y	N	provize z ceny

Tabulka 23: Order_Items

Flags – tabulka názvu typu produktů(novinka, akce, nejprodávanější, doporučené)

Atribut	Datový typ	Klíč	Null	Index	Popis
id	int(11)	PK	N	Y	id typu
name	varchar(255)	N	N	N	název typu

Tabulka 24: Flags

Flag_Products – tabulka typu produktů

Atribut	Datový typ	Klíč	Null	Index	Popis
id	int(11)	PK	N	Y	id typu
product_id	int(11)	FK	N	Y	id produktu
flag_id	int(11)	FK	N	Y	id názvu typu
floor_id	int(11)	N	N	Y	id patra
is_revised	boolean	N	N	N	1 – přidáný, nezměněný, aktualizovaný 0 – můžu smazat

Tabulka 25: Flag_Products

Delayed_Jobs – tabulka pro úlohy puštěné na pozadí

Atribut	Datový typ	Klíč	Null	Index	Popis
id	int(11)	PK	N	Y	id úlohy
priority	int(11)	N	N	Y	id prioritě spuštěné úlohy
attempts	int(11)	N	N	N	pokus do úspěšného spuštění
handler	text	N	N	N	YAML enkódovaný řetězec objektu
last_error	text	N	Y	N	příčina selhání
run_at	datetime	N	N	N	čas, kdy spustil proces
locked_at	datetime	N	Y	N	nastaví čas, kdy se pracuje na úloze
failed_at	datetime	N	Y	N	čas, kdy proces selhal
locked_by	varchar(255)	N	Y	N	kým byl proces zamčen
created_at	datetime	N	N	N	datum vytvoření úlohy
updated_at	datetime	N	N	N	datum úpravy úlohy

Tabulka 26: Delayed_Jobs

Group_Variants – tabulka skupin variant(barva, velikost apod.)

Atribut	Datový typ	Klíč	Null	Index	Popis
id	int(11)	PK	N	Y	id skupiny variant
name	varchar(255)	N	N	Y	název varianty

Tabulka 27: Group_Variants

Variants – tabulka jednotlivých variant

Atribut	Datový typ	Klíč	Null	Index	Popis
id	int(11)	PK	N	Y	id varianty
group_variant_id	int(11)	FK	N	Y	id skupiny variant
value	varchar(255)	N	N	N	hodnota varianty
code	varchar(255)	N	Y	N	kód produktu varianty
availability	varchar(255)	N	Y	N	dostupnost ve dnech
price_vat	decimal(10,2)	N	Y	N	cena produktu varianty
regular_price	decimal(10,2)	N	Y	N	obvyklá cena produktu
floor_id	int(11)	N	N	Y	id patra
is_revised	boolean	N	N	N	1 – přidáný, nezměněný, aktualizovaný 0 – můžu smazat

Tabulka 28: Variants

Product_Variants – vazební tabulka mezi variantami a produkty

Atribut	Datový typ	Klíč	Null	Index	Popis
id	int(11)	PK	N	Y	id
variant_id	int(11)	FK	N	Y	id varianty
product_id	int(11)	FK	N	Y	id produktu
floor_id	int(11)	N	N	Y	id patra
is_revised	boolean	N	N	N	1 – přidáný, nezměněný, aktualizovaný 0 – můžu smazat

Tabulka 29: Product_Variants